PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :
G11B 5/09, 20/14, 20/16

A1

(11) International Publication Number: WO 94/18670

(43) International Publication Date: 18 August 1994 (18.08.94)

(21) International Application Number: PCT/US94/01084

(22) International Filing Date: 31 January 1994 (31.01.94)

(30) Priority Data:
08/012,266 1 February 1993 (01.02.93) US

(71) Applicant: CIRRUS LOGIC, INC. [US/US]; 3100 W. Warren Avenue, Fremont, CA 94538 (US).

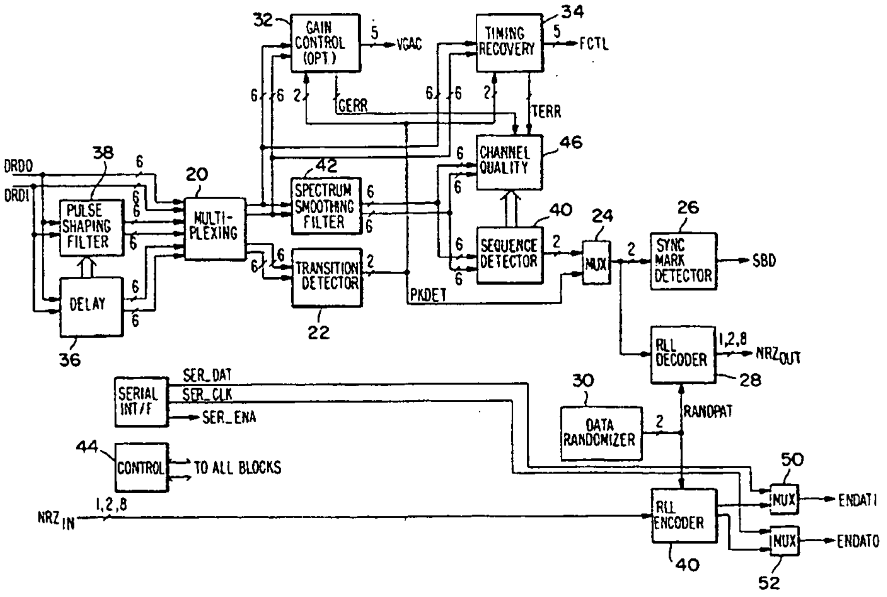
(72) Inventors: BEHRENS, Richard, T.; 659 West Owl Drive, Louisville, CO 80027 (US). ANDERSON, Kent, D.; 3357 W. 97th Avenue, Westminster, CO 80030 (US). ARMSTRONG, Alan; 2312 22nd Avenue, Longmont, CO 80503 (US). DUDLEY, Trent; 5348 South Fox Street, #101, Littleton, CO 80120 (US). FOLAND, Bill; 7793 Emerald Peak, Littleton, CO 80127 (US). GLOVER, Neal; 17 Amesbury, Broomfield, CO 80038 (US). KING, Larry; 397 Ord Drive, Boulder, CO 80303 (US).

(74) Agents: BLAKELY, Roger, W., Jr. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States: AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, LV, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: SYNCHRONOUS READ CHANNEL



(57) Abstract

A synchronous read channel having a single chip integrated circuit digital portion which provides digital gain control (32), timing recovery (34), sequence detection (40), RLL (1, 7) encoding (48) and RLL (1, 7) decoding (28), and channel quality measurement (46) is disclosed. The integrated circuit accommodates both center sampling and side sampling, and has a high degree of programmability of various pulse shaping and recovery parameters and the ability to provide decoded data using sequence detection (40). These characteristics, together with error-tolerant sync mark detection (26) and the ability to recover data when the sync mark is obliterated, allow a wide variety of retry and recovery strategies to maximize the possibility of data recovery.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SI	Slovenia
SK	Slovakia
SN	Senegal
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

SYNCHRONOUS READ CHANNEL

BACKGROUND OF THE INVENTION

In the storage or transmission of digital information, the bits or symbols of the user data are actually transmitted or stored via a physical media or mechanism whose responses are essentially analog in nature. The analog write or transmit signal going into the storage/transmission media or channel is typically modulated by channel bits (typically run-length limited or RLL bits) that are an encoded version of the original user-data bits (non-return-to-zero or NRZ bits). The analog read or receive signal coming from the media is demodulated to detect or extract estimated channel bits, which are then decoded into estimated user-data bits. Ideally, the estimated user-data bits would be an identical copy of the original user-data bits. In practice, they can be corrupted by distortion, timing variations, noise and flaws in the media and in the write/transmit and read/receive channels.

The process of demodulating the analog read signal into a stream of estimated user-data bits can be implemented digitally. Digital demodulation in magnetic mass storage systems requires that the analog read signal be sampled at a rate that is on the order of the channel-bit rate. Maximum-likelihood (ML) demodulation is a process of constructing a best estimate of the channel bits that were written based on digitized samples captured from the analog read signal.

FIGURE 1 shows an exemplary read signal 100, which is a positive-going pulse generated by an inductive read head, for example, from a single media transition such as transition 103 from North-South to South-North magnetization of track 104 on a rotating disk. Typically, the write signal modulates a transition in the state of the media to write a channel bit of 1 and modulates the absence of a media transition to write a 0 channel bit. Thus, transition 103 corresponds to a single channel bit of value 1 in a stream of 0's.

It is common to use run-length-limited (RLL) encoding of the original user data bits, which are arbitrary or unconstrained, into an RLL-encoded stream of channel bits. It may be desirable that there be no less than d zeroes between ones; that is, that the media transitions be spaced by at least d+1 channel bit times. This constraint can help keep to a manageable level the interference effects among the pulses in the analog read signal. On the other hand, because media transitions provide timing information that must be extracted from the read signal to ensure synchronization of the demodulator with the pulses in the read signal, it may be desirable that there be no more than k zeroes between ones; that is, that there be a media transition at least every k'th channel bit time. An RLL(d,k) code is a code that can encode an arbitrary stream of original user-data bits into a stream of channel bits such that the encoded channel bit stream satisfies these two constraints. An RLL code has a theoretical capacity which limits the number of user bits which can be represented in a given number of RLL bits. The capacity is a function of the d and k constraints with d=0 and k=infinite being the limiting (unconstrained) case with a capacity of exactly one. The capacity of an RLL (1,7) code for example is just slightly greater than 2/3 and is exactly 2/3 for any practical implementation, meaning that every pair of user bits will map to exactly three RLL bits.

FIGURE 1, sample set 101 shows the values of four samples in the case of side sampling of read signal 100; i.e. 0.333, 1.0, 1.0, and 0.333. Sample set 101 is equivalent to the set 1, 3, 3, 1; that is, only the ratios among samples are significant. A signal model gives rise to an expected sample sequence for a single or isolated transition in media state. Typically, only a few samples of an isolated media transition are non-zero; in this case, four are non-zero. In a side-sampled signal model such as 1, 3, 3, 1, timing circuitry in the demodulator attempts to maintain a lock on the incoming signal such that two adjacent samples on opposite sides of the peak of an isolated pulse have equal amplitudes and samples are taken at roughly equal time intervals, each a single channel bit time. Synchronization of the samples with the spacing of the bits written on the media is maintained by a timing recovery loop which is in essence a phase-locked loop. Other sample timing arrangements may be useful. In center sampling, the timing circuitry tries to lock the sample times to the read signal pulses such that one sample occurs at the peak of each pulse. Sample set 102 shows the values of four samples in the case of center sampling of a similar read signal 104; i.e., 0.5, 1.0, 0.5, and 0.0 (or 1.0, 2.0, 1.0 and 0.0 depending on the arbitrary normalization used). An expected sample sequence of 1, 2, 1, 0 corresponds to the signal model known in the prior art as Extended Partial-Response Class IV (EPR4). Such sample sequences are samples of a continuous-time analog read-signal waveform such as may be produced in the readback circuitry of a magnetic storage device. For a system that is bandwidth limited to 1/(2T), where T is the sample spacing in time, the sampling theorem declares that the continuous time waveform must be superposition of sinc functions (sinc(x) is defined as sin(x)/x for x<>0, and as 1 for x=0), with one sinc function centered at each sample point and of amplitude equal to that sample value and with zero crossings at all other sample points. As an example, in saturation magnetic recording, the current in an inductive write head takes on values of +1 and -1. The basic excitation applied to the recording channel is a step in current from +1 to -1, vice versa, in the analog write signal. This step in write current produces a transition in the magnetization state of the media as it moves past the head. When an inductive read head is passed over this magnetic media transition, a voltage pulse is induced by the bandwidth limited differentiating interaction of the head with the magnetization of the media. By suitable filtering or equalization, the sequence of samples on an isolated transition response pulse can be made to {..., 0, 0, 1, 2, 1, 0, 0, ...}, in which case the recording or transmission channel matches the EPR4 signal model. Another sample sequence well known in the prior art is the Partial Response Class IV signal model (PR4), which corresponds to an expected sample sequence of 0, 1, 1, 0. Further, as one is designing or taking measurements on a write/media/read channel, it may be desirable to take into account the exact response, noise and distortion characteristics of the channel in selecting the signal model to be implemented in the demodulator. Thus, there is a need for a demodulator that is programmable as to the signal model, or expected sequence of sample values for an isolated media transition. In situations such as mass information storage in magnetic media, significant storage-system speed and capacity gains can be realized if the information bits can be closer together in position/time on the media. further, as media transitions are more closely positioned, the writing and reading processes become more sensitive to the distortion, timing variations and noise that are inevitably introduced in the processes of writing, storing, and reading. Also, as the transitions become closer, the ability of the media to fully transition from, say, North-South magnetization to South-North magnetization may be taxed. Also, as the media transitions become closer, interference effects increase among adjacent or nearby transitions. FIGURE 2 shows how positive-going pulse 200 from first media transition 201 combines with negative-going pulse 202 from second transition 203 to produce analog read signal 204, which can be viewed as the interference of the two pulses. Adjacent media transitions always give rise to read pulses of opposite polarities because they always are created by transitions of opposite types, for example North-South changes to South-North in transition 201, so adjacent transition 202 must be South-North changing back to North-South. Read signal 204 might give rise to a sequence of samples such as 0.333, 1.0, 0.667, -0.667, -1.0, 0.333. To the extent that the read process is linear (and it may not be entirely linear), the voltage waveform induced in the read head will be the superposition of a sequence of pulses, where each pulse is the response to an isolated magnetic transition on the media. Clearly, engineering a high-performance read channel is a complex challenge given the combined effects of the limited sampling rate in a digital demodulator, possibly incomplete transitions in the media, interference among read-signal responses to media transitions, and distortion, timing variations, noise and flaws in the media and in the write and read channels. The prior art uses a method known as partial-response signaling to increase media transition rates. Partial-response signaling is described in the book "Digital Transmission of Information", by Richard E. Blahut, 1990, pp. 139-158 and 249-255. This method allows the analog response of the storage/transmission media and of the write/transmit and read/receive circuitry to a media transition to overlap with the response to adjacent transitions associated with subsequent information bits. If properly implemented, this method can achieve higher information bit rates/densities than the alternative or requiring the media transitions to be spaced such that the read signal responses do not overlap. Such a method requires a sequence detector which can make its decisions not on a bit-by-bit basis but by examining the context of the surrounding read signal.

In a magnetic disk drive, the surface of the magnetic media is logically divided into concentric rings called tracks. The distance around the track varies as a function of the radius at which the track lies. Since it is desirable to keep the rate of revolution of the disk constant to avoid mechanical delays in accelerating and decelerating the disk, it is necessary to either store an amount of data on each track which is proportional to the length of the track (this requires a different data transfer rate for each track) or to vary the physical transition spacing on the media so that pulses are widely separated at the outside diameter and crowded very close at the inner diameter of the recording surface (this is wasteful of the magnetic media which is only sparsely used at the outer diameter). A practice known as zoned recording is a popular compromise between these two extremes. In zoned recording, a group of tracks (a zone) is established in which every track in the zone holds the same amount of data. Thus each zone requires a different data transfer rate, but the number of data transfer rates which need be supported is reduced (more coarsely quantized). This still leaves a variation in the physical spacing of transitions between the inside and outside diameters of each zone resulting in a variation in pulse shape.

Partial-response signaling has just recently been incorporated into mass storage devices and then in a limited form. One prior-art magnetic disk drive using partial-response signaling only supports PR4 (pulses with the samples of ..., 0, 1, 1, 0, ...). PR4 signaling has only very limited inter-symbol interference evidenced by only two non-zero samples in the pulse. To increase the capacity of the media, the user of a PR4 read channel must increase the equalization of the pulses (slim the pulses) in order to limit the inter-symbol interference of adjacent pulses so that any pulse only affects two read signal samples. The increased equalization also enhances the noise accompanying the signal, making the detection task more difficult and errors more likely. U.S. Patent 4,945,538 by Patel covers a similar situation but with EPR4 signaling and an RLL(1,7) code. This improves the allowed amount of inter-symbol interference, increasing it to three non-zero samples of (..., 0, 1/2, 1, 1/2, 0, ...). Both of these techniques will allow an increase in capacity but are limited in the variety of pulse shapes which can be detected and therefore limited by how much equalization (pulse slimming) may be performed before the effect of equalizing the noise (noise enhancement) becomes intolerable.

Thus, there is a need for a flexible read channel which can accommodate a wide variety of pulse shapes as will be seen in each zone. There is also a need to allow larger amounts of controlled inter-symbol interference between pulses (pulses with more than two or three non-zero pulses) in order to continue increasing the capacity of the recording media.

SUMMARY OF THE INVENTION

A synchronous read channel having a single chip integrated circuit digital portion which provides digital gain control, timing recovery, equalization, digital peak detection, sequence detection, RLL(1,7) encoding and decoding, error-tolerant synchronization and channel quality measurement is disclosed. The integrated circuit accommodates both center sampling and side sampling, and has a high degree of programmability of various pulse shaping and recovery parameters and the ability to provide decoded data using sequence detection or digital peak detection. These characteristics, together with the error-tolerant sync mark detection and the ability to recover data when the sync mark is obliterated, allow a wide variety of retry and recovery strategies to maximize the possibility of data recovery. Various embodiments, including an embodiment incorporating the analog functions as well as the primary digital functions of the read channel in a single integrated circuit, and preferred embodiments utilizing a reduced complexity, programmable modified Viterbi detector supporting a broad class of partial response channels are disclosed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 shows a state transition on a medium such as a track of a disk drive and its associated pulse in an analog read signal. It also shows two digitized sample models of such read-signal pulses.

FIGURE 2 shows two adjacent medium transitions and their individual and combined read-signal pulses.

Figure 3 is an overall block diagram of the present invention.

Figure 4 is a block diagram illustrating the details of the gain control circuit 32 of Figure 3.

Figure 5 is a block diagram illustrating the details of the timing recovery circuit 34 of Figure 3.

Figure 6 is a block diagram illustrating the details of the spectrum smoothing filter 42 of Figure 3.

DETAILED DESCRIPTION OF THE INVENTION

The CL-SH4400 is a specific embodiment of the present invention designed to work with a companion analog integrated circuit and a disk controller to form a state of the art high density magnetic disk drive. In that regard, the uniqueness of the present invention, while used in a digital read-write channel, is primarily related to its read capability and versatility.

The companion integrated circuit with which the CL-SH4400 is specifically intended to operate implements a VGA (Variable Gain Amplifier), a tunable analog filter, an analog to digital converter, a timing VFO (Variable Frequency Oscillator), write pre-compensation and servo demodulation functions. Accordingly, in a read operation, the CL-SH4400 does not receive an analog signal but instead receives already digitized read information in the form of digitized analog read channel samples. Further, while the timing Variable Frequency Oscillator and the Variable Gain Amplifier are on the companion integrated circuit and are not part of the present invention, the timing VFO and the Variable Gain Amplifier are each digitally controlled through digital control signals generated in the CL-SH4400. Accordingly, in the specific embodiment to be described, digital control feedback signals for both the VFO and the VGA are generated in the CL-SH4400 even though the control loops for the timing recovery and the automatic gain control functions are actually closed within the companion analog integrated circuit. In that regard, it should be particularly noted that the automatic gain control signal may alternatively be generated on the analog companion integrated circuit, as the same may be readily generated in the analog domain rather than in the digital domain. Accordingly, particularly the generation of the digital gain control to be described as part of the CL-SH4400 is an optional design choice readily relegated to the companion integrated circuit if desired.

Figure 3 provides a block diagram illustrating the general organization of the CL-SH4400. As may be seen in Figure 3, the digitized read data for the CL-SH4400 is provided in an N-bit parallel form as digitized read data DRD0 and DRD1. Each of these two signals in the preferred embodiment disclosed is a 6-bit digitized read data signal. These two N-bit signals represent digitized samples of a read signal directly from a read head of the storage device after analog amplification and analog filtering. Those skilled in the art will recognize that the purpose of the analog amplifier and the analog filter is to scale the signals to the input range of the digital to analog converter and to attenuate frequencies above the Nyquist frequency (1/2 the sample frequency) to avoid signal distortion due to aliasing. In general, the analog filter will perform pulse shaping as well. The digitized read data signal DRD0 is a digitized read signal sample effectively taken near the center of a channel bit time (defined by the VFO frequency), subject however to a small amount of timing error or intentional timing set point offset in the VFO. The digitized read data signal DRD1 is the corresponding digitized read sample effectively taken near the center time of the previous logical channel bit, subject of course to similar timing errors and timing set point offsets. These two digitized read data signals are processed in the CL-SH4400 in a parallel or simultaneous manner so that ultimately in the CL-SH4400, two successive bits of digital read data will be derived from one set of DRD0 and DRD1 signals which together with successive bit pairs are decoded by a run-length limited (RLL) decoder and derandomized if applicable (e.g. if initially randomized) to provide the NRZ data output stream of the device. The processing of two digitized read data samples simultaneously doubles the throughput of the CL-SH4400 for a given clock rate without doubling the circuitry required, particularly in the sequence detector, though the present invention is not specifically limited to processing of two digitized read data sample at a time. One could process one digitized read data sample at a time, or alternatively process more than two digitized read data samples at a time, if desired. In that regard, the number of N-bit digitized read data sample connections to the chip normally will equal the number of samples processed at a time, though such signals could be multiplexed so that the number of N-bit digitized read data samples connections to the chip is less than the number of samples processed together.

In one mode of operation, multiplexer 28 couples the DRD0 and DRD1 signals directly to a transition detector 22 which processes the successive samples to detect the presence of transitions in each of the two digitized read data signals. The output of the transition detector is a low or high level during the respective bit times (delayed as described in the co-pending application) depending upon whether a transition (providing a high level output) or no transition (providing a low level output) was detected. The output of the transition detector 22, the peak detected signal PKDET, in this mode would be coupled to multiplexer 24 and through a sync mark detector 26 to provide a sync byte detected output SBD if a sync byte was in fact detected, and to couple the two bits to the RLL decoder 28 which decodes the bit stream to provide the NRZ data out digital data. In the preferred embodiment run length constraint violations are detected and optionally multiplexed onto the NRZ data out lines. These may be used by an error correcting system within the disk controller. In the preferred embodiment, an error tolerant sync mark detector 26 is used. This detector is designed to achieve a level of error tolerance for the synchronization function equal to that achieved for the data field by the error-correction code implemented in the disk controller. This is achieved in part by employing an error-tolerant Synchronization Mark pattern for minimum cross-correlation with the preamble and for minimum auto-correlation, and by making the number of four-channel-bit groups which must be detected programmable. The synchronization mark recovery procedure may be used to recover data when a severe defect has destroyed the entire synchronization mark. When using this mode, the CL-SH4400 first goes through a normal timing and gain acquisition procedure while counting channel bits. The synchronization mark is assumed to have been detected when the count matches the synchronization mark recovery count. By varying the synchronization mark recount, the microcontroller can vary the assumed starting point of a header or data area until the correct starting point is tried, whereupon the sector will be recovered if there is no other error beyond the capability of the error correction code in the disk controller.

In the CL-SH4400, the NRZ data output is user selectable as a serial bit stream, a 2-bit parallel stream or character wide (8-bit wide) digital data. If the data was randomized prior to storage, the data randomizer 30 may be enabled to de-randomize the data from the RLL decoder 28 before being provided as the NRZ data output.

The output of the transition detector 22 in this mode is also provided to the gain control circuit 32 and the timing recovery circuit 34. Also the N-bit digitized samples DRD0 and DRD1 are coupled through multiplexer 20 to the gain control circuit 22 and the timing recovery circuit 34. The gain control circuit 32 is shown in more detail in Figure 4. The gain errors (the difference between the programmable desired signal level referred to as the Gain Set point, and each digitized data signal) are determined for each digitized read data sample by the gain error circuit 33. The outputs PKDET of the transition detector 22 provide references to control the multiplexer 35 of the gain control circuit 32, as the gain adjustments are determined by the signal amplitudes of transitions and not the signal levels between transitions. The automatic gain control signal VGAC (5-bits in the preferred embodiment) for coupling back to the companion integrated circuit for analog amplifier gain control is provided by the digital gain loop filter 37. The gain loop filter includes a loop filter coefficient which is independently programmable for tracking and acquisition. The individual gain errors are also coupled to a channel quality circuit 46 as the signals GERR so that gain control performance can be measured to determine the best choice of loop filter coefficients and other parameters of the channel with respect to the performance of the automatic gain control loop.

The timing recovery circuit 34 in the CL-SH4400 is shown in greater detail in Figure 5. Timing recovery and maintenance of synchronization, of course, can only be done upon the detection of a transition, as the absence of transitions contains no timing information. The timing recovery circuit controls the read clocks which are synchronized to the read wave form. In the timing recovery circuit, a phase detector 39 digitally computes the phase error in the sampling instants of the analog to digital converter on the companion chip from the digitized sample values during transitions, as indicated by the signals PKDET. Providing timing error corrections only at transition times reduces the noise (jitter) in the timing loop. The sequence of measured phase errors is digitally filtered by filter 41 to produce a frequency control signal which is fed back to the companion chip, in the preferred embodiment as the 5-bit frequency control signal FCTL.

The timing recovery circuit has two modes of operation, acquisition and tracking. The appropriate range and resolution of the frequency control signal FCTL to the analog companion part is not the same for the two modes. In the acquisition mode, the necessary frequency control range is larger (wider range of possible frequency settings) than in tracking. Conversely, in tracking the required resolution (minimum step of frequency) is finer. To meet these conflicting requirements without unduly increasing the number of bits in the FCTL interface, the resolution and range are made to depend on the mode of operation, and a signal ACQ is used to communicate to the analog companion part which mode of operation is being used. When the operating mode is switched from acquisition to tracking, the last frequency setting during acquisition is stored in the companion analog part and the value on the FCTL bus during tracking is taken as an offset from the stored setting. In the preferred embodiment (CL-SH4400) the range and resolution are decreased by a factor of 8 between acquisition and tracking.

The timing recovery circuit 34 also includes a programmable timing set point. The timing set point permits a wider range of sampling strategies which enables the support of a wider range of pulse shapes. The timing set point is useful on retry in the event of the detection of an uncorrectable error. Also the digital filter includes two coefficients which are independently programmable for acquisition and tracking. These are also usable in a retry strategy to change the bandwidth and hence the response time of the timing loop. Like the individual gain errors GERR, the individual timing errors TERR are also coupled to the channel quality circuit for contribution to the quantitative analysis of the channel quality with respect to timing recovery. One alternate embodiment of the timing recovery block includes a frequency error detector. The frequency error detector is used to decrease the time required for the timing loop to lock to the channel bit frequency and phase during the acquisition period before encountering data.

As variations on the mode of operation just described, the two N-bit digitized read data signals DRD0 and DRD1 may be passed through a pulse shaping filter 38 prior to being coupled to multiplexing block 20. The pulse shaping filter provides digital filtering at the cost of a small amount of delay with two user selectable coefficients PC1 and PC2, independently programmable in the filter structure. This pulse shaping filter, of course, is in addition to any filtering done in the analog domain, and is an example of the flexibility and adaptability of the present invention. In particular, the effect of the pulse shaping filter may be eliminated by multiplexing block 20, or alternatively, the pulse shaping filter may be used with coefficients user selected, and thus variable, to provide the best performance of the overall storage system read channel with the flexibility to accommodate changes in pulse shape when changing from one recording zone to another, and to allow coefficient variations as part of overall device parameter variations for systematic retries upon the detection of uncorrectable errors in the subsequent error detection and correction (EDAC) operations. The pulse shaping filter of the preferred embodiment is a finite impulse response digital filter which means that the output is a function of the current and past inputs but not a function of its own past outputs. The delays necessary for the filter to remember the past inputs are shared by delay 36 to provide a separate delay path to multiplexing block 20. The delay path through delay 36 provides an amount of delay equivalent to the delay of pulse shaping filter 38. Multiplexing block 20 is provided to give a maximum of flexibility in modes of usage, by providing a separate source of input for the transition detector and the group of blocks comprising the sequence detector 40 (by way of spectrum smoothing filter 42), the gain control circuitry 32 and the timing recovery circuitry 34. The multiplexing block is designed so that all blocks may operate upon the raw input samples provided by DRD0 and DRD1, or alternatively the pulse shaping filter may be placed in one of the multiplexing block's output paths with the delay placed in the other path. The delay is necessary in this case so that the transition detector's outputs are synchronized with the sample values reaching the gain control circuitry and timing recovery circuitry. Finally, the pulse shaping filter may feed both of the multiplexing block's output paths. In general the multiplexing block could be designed so that by programming the multiplexing block each block at the multiplexing block's outputs (transition detector 22, gain control 32, timing recovery 34, and sequence detector 40 by way of spectrum smoothing filter 42) could receive raw input samples, delayed raw input samples, or filtered input samples independent of the data received by the other blocks. In the preferred embodiment of the present invention however, the gain control circuitry and the timing recovery circuitry have the capability of compensating for the pulse shape in an effort to increase the accuracy of the gain and timing recovery loops, and to reduce the amount of circuitry, the gain control circuitry and timing recovery circuitry assume the same pulse shape for which sequence detector 40 is programmed. This basic pulse shape received by the sequence detector is only marginally affected by the addition of the spectrum smoothing filter, which is designed to reduce only the head bumps at the tails.of the pulse and does not seriously affect the center of the pulse except to correct for head bumps due to neighboring pulses. In that regard, the present invention includes a channel quality circuit 46 for measuring the quality of the read channel as earlier described. This provides not only quantitative channel evaluation, but in addition allows selection of read channel parameters such as, but not limited to, the coefficients PC1 and PC2 in the pulse shaping filter to best adapt the read channel to the characteristics of the storage medium and the pulse form and characteristics being read therefrom.

The present invention further includes a sequence detector 40 which receives as its input the two N-bit digital read data signals DRD0 and DRD1 as may be modified by the pulse shaping filter 38 and as may be additionally modified by the spectrum smoothing filter 42. In that regard, the spectrum smoothing filter 42, as shown in Figure 6, contains two delays FD1 and FD2 and four coefficients SC1, SC2, SC3 and SC4, all of which are independently programmable. The delays may be programmed from 0 to 23 channel bit intervals. The entire spectrum smoothing filter, or just its precursor correcting portion 43, can be disabled. The spectrum smoothing filter is designed to reduce the undershoots from the finite pole tips of a thin film head, or to reduce the bumps from the secondary gap of a single or double-sided MIG head. In the frequency domain, the filter acts to smooth out undulations caused by head bumps. If the precursor is disabled, the delay of the filter is disabled, whereas if a head which is not subject to head bumps is used, the post-cursor may be disabled.

The pulse shaping filter and the spectrum smoothing filter together form a digital equalizer which can modify the equalization done in the analog filter of the companion integrated circuit, and along with the companion integrated circuit provides support for changing equalization needs from head to head and zone to zone of the magnetic storage device. The parameters for the pulse shaping filter and the spectrum smoothing filter are loaded and/or varied by microcontroller 44 on initialization and during head seeks.

The sequence detector 40 is a partial response sequence detector. This allows the analog response of the read channel to a storage medium transition to overlap with the response to adjacent transitions associated with subsequent information bits. In comparison to most prior art read channels for magnetic storage media, the use of a partial response detector allows higher information storage densities in comparison to the prior art alternative of requiring the medium transitions to be sufficiently spaced from each other so that the read signal responses do not overlap significantly, thereby allowing each transition to be individually detected irrespective of the nearest neighboring transitions.

The particular sequence detector used in the present invention is a uniquely modified form of Viterbi detector which substantially preserves the full performance of the Viterbi algorithm in a substantially reduced complexity sequence detector. The basic Viterbi algorithm is described in the book "Fast Algorithms for Digital Signal Processing" by Richard E. Blahut, 1985, pages 387-399. In accordance with the Viterbi algorithm, a Viterbi detector does not attempt to decide whether a medium transition has occurred immediately upon receipt of the read sample or samples that correspond to that transition. Rather, as samples are taken from the read signal, the Viterbi detector keeps a running tally of the error between the actual sample sequence and the sample sequence that would be expected if the medium had been written with a particular sequence of transitions. Such an error tally is simultaneously kept for several possible transition sequences. As more samples are taken, less likely choices for transition sequences are pruned from consideration. If a set of possible sequences of medium transitions is appropriately constrained, then the location of each medium transition becomes known with a high degree of likelihood within a reasonable time after taking the samples corresponding to that transition. Because of the time delay between a sample acquisition and the determination of whether that sample represented a transition or an absence of a transition, the gain control circuit 32 and the timing recovery circuit 34 are both still referenced to the output of the transition detector 22, which has a more immediate response to the occurrence of a transition. Also, while in general the output of the sequence detector, when used, should be more accurate in ultimately determining whether a transition occurred at a particular bit time, an error in the output of the peak detector will have little effect on gain and timing. Specifically, failure to detect a transition will only slightly delay gain control and timing error corrections, and an isolated false detection of a transition will only slightly perturb the gain control and timing accuracy. This should be more than made up by the increased accuracy of the bit stream detection by the sequence detector's consideration of what comes before and after a particular digitized read data sample.

If the present invention is realized in an embodiment wherein digitized read data is processed a single bit time's worth at a time, a Viterbi detector of a conventional design may be used, or if two or more bit time's worth of samples are to be processed simultaneously, as in the preferred embodiment of the present invention, a conventional Viterbi detector could be modified for that purpose.

In a typical Viterbi detector implemented using the ADD, COMPARE, SELECT (ACS) method, each state in the expected sample sequence model is associated with a hardware module to perform the functions of adding new branch error metrics to path error metrics, comparing path error metrics, and selecting the path having the lowest path metric. In the sequence detector used in the preferred embodiment, an ACS module may have two or more sequence model states dynamically associated with it such that at some times, one sequence model state is associated with it, and at other times, another sequence model is associated with it. This reduces the number of ACS modules required and also reduces the size and complexity of the detector path memories which must store one path for each ACS module. Groups of sequence model states may be chosen to share an ACS module without significant loss in performance as compared to the conventional Viterbi detector. These detectors support a wide range of sample models by making the expected sample sequence of an isolated medium transition programmable through control 44. By way of specific example, the sequence detector used in the CL-SH4400 disclosed herein will support the PR4, EPR4 and EEPR4 sample models, among others. In addition, the alternating polarity of pulses is enforced, as is a minimum run length constraint of d=1.

The d=1 constraint in the RLL(d,k) coding is an important constraint in the present invention, especially for applications where the storage system uses thin-film magnetic media. For thin-film magnetic media, there is an effect known as partial erasure which puts a practical limit on how close two magnetic transitions may be written. The effect is due to a ragged (or zig-zag) boundary between regions of opposite magnetization. As the transitions become too close, the zig-zags begin to overlap and the area of opposite magnetic polarity between two transitions starts to disappear. The result is that as the read head flies over the partially erased transitions, the amplitude of the corresponding read signal pulses is diminished. This is a non-linear, data pattern dependent effect which is difficult to compensate for. The d=1 constraint remedies this situation by preventing magnetic transitions in two consecutive channel bit times. The drawback is that the d=0 constrained code may typically represent 8 NRZ bits with 9 RLL bits (rate 8/9) while the d=1 constrained code can only represent 6 NRZ bits with 9 RLL bits (rate 2/3). For example, to store 8 NRZ bits, the d=0 constrained code will store 9 channel bits while the d=1 constrained code will store 12 channel bits in the same amount of space, hence the d=1 channel bit interval is 3/4 the size of the d=0 channel bit interval. Fortunately, the magnetic transitions have a minimum spacing of 2 channel bits and therefore the minimum distance between two transitions has increased by 3/2 with respect to the corresponding d=0 constrained code. This makes the d=1 constrained read channel a good solution for increasing storage capacity in applications where the minimum transition spacing is close enough for partial erasure effects to be noticeable.

The sequence detector utilized in the CL-SH4400 can be programmed to operate on any channel response which can be well represented by sequences in the form of a, b, 1, c wherein the selection of a, b and c allow the ability to accommodate pulse asymmetry which might otherwise require that the read signal pass through an analog or digital phase equalizer prior to entering the sequence detector. The levels a, b and c also give the ability to select between center and side sampling. Center-sampled pulses are notably those for which the sample levels a, b, 1, and c are selected such that 1 is very near the peak of the pulse, b and c are roughly halfway down their respective sides of the pulse, and a is near zero, for example, the sample levels 0, 1/2, 1 and 1/2. Side-sampled pulses are notably those for which the sample levels are selected such that 1 and b (which is about 1) straddle the peak of the pulse, for example the sample levels 5/16, 1, 1 and 5/16. This choice of side versus center sampling also affects the manner in which gain error and phase error are calculated in the gain control loop and timing recovery loop. The option to choose between side and center sampling allows the user a wider range of possible trade-offs between the amount of equalization (filtering) used to shape the raw pulse shape into the target pulse shape of the sequence detector and the amount of noise enhancement which arises as a consequence of shaping the raw pulse. Hence the read channel can be more suitably matched to the storage medium to provide better performance.

Referring again to Figure 3, for writing information to the storage medium, the NRZ input data is provided to a run length limited encoder 48, in the CL-SH4400 through a user selectable serial line, a two bit parallel form or an eight bit byte parallel form. The run length limited encoder provides the desired run length limited coding, in the preferred embodiment an RLL (1,7) coding, randomized before encoding or not, depending upon the enabling of the data randomizer 30, with the encoded data being provided to multiplexer 52, in the preferred embodiment in a two-bit wide form. In the case of writing to the storage medium, the serial enable signal SER_ENA will be deasserted so that the multiplexer 50 and 52 will provide the encoded data bits 1 (the most significant of the two parallel data bits) and 0 (the least significant of the two parallel data bits) to the companion integrated circuit to write the same to the storage medium.

When not writing, the serial enable signal SER_ENA may be asserted, at which time multiplexers 50 and 52 are switched so that serial control address and data may be transferred on the SER_DAT line to the companion integrated circuit synchronous with the associated serial clock signal SER_CLK as the output signals of the two multiplexers 50 and 52. Multiplexing of these two chip pins of the single chip CL-SH4400 integrated circuit helps reduce the pin count without loss of performance or flexibility. This serial interface is provided to eliminate the need for the companion integrated circuit to interface with the bus of the microprocessor and eliminates a potential coupling between the noisy microprocessor bus and in the sensitive analog circuitry of the read channel in the companion integrated circuit. This also provides a benefit in pin count since the microprocessor bus interface would require numerous additional pins on the companion integrated circuit. Each of the control registers of the companion integrated circuit are mapped to corresponding register addresses in the integrated circuit of the present invention, when one of these registers is written to, the serial interface initiates a serial transfer write operation, sending the data to the appropriate register in the companion integrated circuit. The preferred embodiment of the present invention includes two modes, one in which a status bit which can be read to determine whether or not the serial transfer write operation is complete and another in which the integrated circuit of the preferred embodiment forces the microprocessor to pause while the serial transfer write operation is in progress. Similarly, a read of a register in the companion integrated circuit is performed by reading the corresponding register in the integrated circuit of the present invention. This initiates a serial transfer read operation. In the first of two serial interface read modes, the integrated circuit of the preferred embodiment will signal the microprocessor to pause until the serial transfer read operation is complete at which time the serially transferred data will be accessible at the pins of the integrated circuit of the present invention. In a second mode, the integrated circuit of the preferred embodiment will return the data left over from the previous serial transfer read operation, and once the current read operation is completed, it will initiate a new serial interface read operation with the address just supplied. The preferred embodiment includes a status bit which can be read to determine whether or not the new serial transfer read operation is complete. Once complete, the microprocessor may initiate a second read operation to retrieve the data originally desired and to initiate another serial transfer read operation at a new address for future use if desired. Summarizing the two modes, in one mode the microprocessor is made to wait, in the second mode the microprocessor must read the register twice, once to supply the register address and second time to retrieve the data and possibly supply the next register address.

Another embodiment of the present invention, namely part number CL-SH3300 integrated circuit, incorporates the essential functions of the CL-SH4400 and the companion integrated circuit in a single integrated circuit.

In the embodiments hereinbefore described, numerous parameters were described as being programmable, and as such, as being useful for varying on retries in the event of the detection or repeated detection of an uncorrectable error. Obviously additional programmable or fixed parameters may also be incorporated, such as by way of example, additional filter coefficients in the pulse shaping and other filters. It should be noted that particularly certain parameters, such as the parameters of the pulse shaping filter 38 and the spectrum smoothing filter 42, may be made adaptive, or a combination of adaptive and programmable. By way of example, coefficients may be made adaptive, while the time constants of the adaptive characteristics and perhaps offsets, wave shapes, asymmetries and compensation for nonlinearities are made programmable. In that regard, it should be noted that given an uncorrectable or repeated uncorrectable error, there is no harm in further attempts at a successful read with different parameters, side versus center sampling, filtering versus different, less or no filtering, etc. Further, normally one would use the sequence detector on read for its superior detection capabilities over a peak detector. However in the case of hard errors wherein the errors in the output of the sequence detector exceed the error correction capability of the EDAC code used, a further retry strategy may include switching to an output derived from the peak detector, as previously described, rather than the sequence detector. This will add noise errors characteristic of transition detection effectively within the single bit-time of the possible transition, but will eliminate whatever additional propagation of the hard errors (for example media defects) may be caused by the sequence detector.

The effect of the noise errors may be substantially eliminated by multiple reads, followed by a majority vote to cancel all or most of the effect of the noise errors, yielding an opportunity for a successful error correction when the same could not be achieved with the sequence detector in the path. Here again, pulse shaping filter characteristics may be varied and multiple retries of the peak detector multiple reads executed, as reasonable exhaustion of all opportunity for a successful read is better than a fatal error.

There have been disclosed and described herein preferred and alternate embodiments of a new and unique synchronous read channel which include a sequence detector with a flexible architecture capable of implementing a broad range of partial response polynomials. While an embodiment of the present invention which supports only one or two partial response channels would be highly useful, the detector used in the preferred embodiment of the present invention supports a broad class of partial response channels, including but not limited to PR4 (1,7), EPR4 (1,7) and EEPR (1,7). While the sequence detector is normally operative upon a read, the output of a digital peak detector may be enabled as the output of the read channel if desired. These and other inventive features of the invention will be apparent from the preceding description.

Thus while a preferred and alternate embodiments of the present invention has been disclosed and described in detail herein, it will be obvious to those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope thereof.

CL-SH4400
Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

Sampled Amplitude Digital R/W Channel

FEATURES

Channel Flexibility

- The CL-SH4400 provides the digital core of a highly flexible, high-performance sampled-amplitude read/write channel at channel rates up to 96 MHz.

Complete RLL Detector

- Rate 2/3 RLL (1,7) recording code
- Programmable digital equalization
- Programmable SofTarget™ sequence detection
- Digital gain control
- Digital timing recovery

Error Tolerance

- Error-tolerant synchronization
- Channel quality circuitry for error rate testing and filter/detector calibration
- Erasure pointer generation

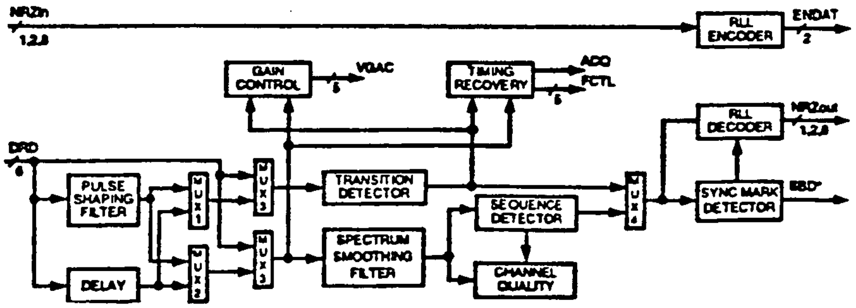
Head Support

- Monolithic, composite, thin-film, MIG and MR

OVERVIEW

The CL-SH4400 Sampled-Amplitude Digital Channel is a VLSI component designed to work with a companion analog part and a Disk Controller to provide the majority of drive and controller hardware necessary to build a state-of-the-art, high-density, magnetic disk drive. The CL-SH4400 implements the digital portion of a sampled-amplitude read/write channel employing advanced partial response polynomials and SofTarget™ sequence detection technology. It supports data rates up to 64 Mb/s.

Functional Block Diagram



CIRRUS LOGIC

CL-SH4400

Sampled Amplitude Digital R/W Channel

The CL-SH4400 Sampled-Amplitude Digital Channel provides the digital core of a highly flexible, high-performance sampled-amplitude read/write channel which provides data rates up to 64 Mb/s. A companion analog part, the VM6400, implements the frequency synthesizer, variable gain amplifier, timing VFO, tunable analog filter, analog-to-digital converter, write precompensation, and servo demodulation functions. The CL-SH4400/VM6400 chip set supports hard-sectored magnetic disk drive applications employing embedded servo techniques and zone bit recording. The functional partition of the CL-SH4400/VM6400 chip set is shown in Figure 1.

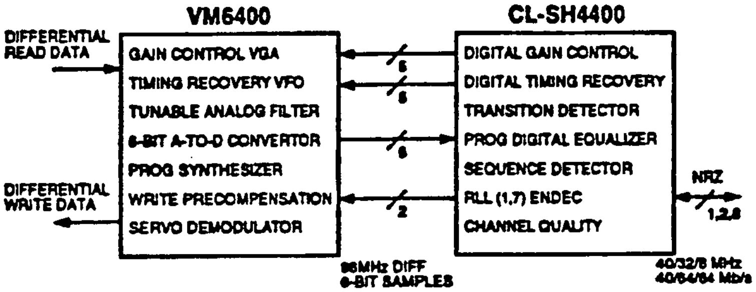


Figure 1 CL-SH4400/VM6400Functional Partition

The CL-SH4400 provides digital gain control, timing recovery, equalization, sequence detection, RLL (1,7) encoding and decoding, error-tolerant synchronization, and channel quality measurement. The high degree of programmability of each of these functions allows the CL-SH4400 to support highly adaptive channels tailored for each drive, head, and zone.

The format of a typical sector containing one embedded servo area is shown in Figure 2. During a read(write) operation, the Disk Controller reads the offset of the servo area from the ID. It deasserts Read(Write) Gate after the specified number of bytes have been processed and reasserts it after a programmed number of bytes equal to the length of the servo area. The VM6400 and external circuitry read and decode the servo bursts and servo data. During a read operation, the CL-SH4400 freezes its gain control and timing recovery states while Read Gate is deasserted.

PREAMBLE	SYNC	ID	CRC	SPLICE	PREAMBLE	SYNC	DATA	SERVO	SPLICE	PREAMBLE	SYNC	DATA	SPLICE

Figure 2 Typical Sector Format

CL-SH4400
Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

ADVANTAGES

Benefits

- Flexible architecture implements broad range of partial response polynomials.
- Provides significant implementation and performance benefits over conventional peak detection.
- RLL (1,7) code reduces nonlinear effects associated with closely-spaced transitions.
- Programmable equalization, sampling, and detection alternatives provide optimum match to channel characteristics and permit implementation of adaptive channels.
- SofTarget™ sequence detection accommodates a lower signal-to-noise ratio than hard thresholding of individual samples and allows significant intersymbol interference without negative consequences.
- Detector supports a broad class of partial response channels including PR4 (1,7), EPR4 (1,7) and EEPR4 (1,7).
- Error tolerance features support the higher soft error rates and higher defect densities encountered at higher recording densities.
- One/Two/Eight-bit NRZ Data interface connects to a broad range of Disk Controllers.
- Intelligent power management minimizes operational and idle power consumption.

Key Features

- One-bit, two-bit, or eight-bit NRZ Data interface
- Rate 2/3 RLL (1,7) recording code
- Digital Gain Control
- Digital Timing Recovery
- Digital Equalizer
- SofTarget™ Sequence Detector
- Error-tolerant synchronization
- Programmable gain, timing, equalization, sampling, and detection alternatives
- Channel quality circuit for filter/equalizer/detector calibration and error rate estimation
- Erasure pointer generation
- Support for monolithic, composite, thin-film, MIG and magneto-resistive head technologies.
- Intelligent power management

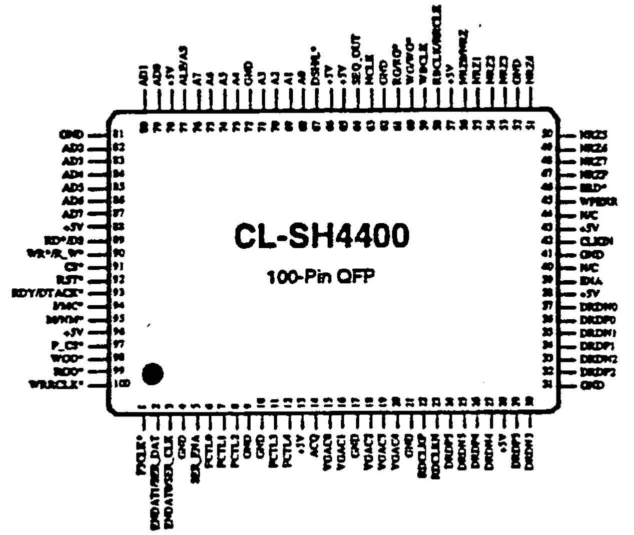
CIRRUS LOGIC

CL-SH4400
Sampled Amplitude Digital R/W Channel

1. PIN INFORMATION

The CL-SH4400 is available in a 100-pin Plastic Quad Flat Pack (PQFP) package. The pin diagram below shows this package.

Pin Diagram for the 100-Pin Plastic Quad Flat Pack (PQFP) Package



2. PIN ASSIGNMENTS

The following conventions are used in the pin assignment tables. Asterisk (*) denotes a negative true signal. (I) indicates an input pin. (O) indicates an output pin. (OD) indicates an open-drain output pin. (I/O) indicates a bidirectional pin. All unused inputs must be tied to the inactive state, V_{CC} or GND.

2.1 Microcontroller Interface Pins

SYMBOL	PIN	TYPE	DESCRIPTION
RST*		I	RESET: Assertion of this signal stops all operations within the CL-SH4400 and deasserts all output control signals. All bidirectional signals are set to the high-impedance state.
I/MC*		I	INTEL/MOTOROLA: This signal selects the microcontroller interface to be used. When this signal is high, it selects the Intel bus control interface. When this signal is low, it selects the Motorola bus control interface. An internal pullup allows this signal to be legally "floated" to select the default Intel bus control interface.
M/NM*		I	MULTIPLEXED/NONMULTIPLEXED: When this signal is high, AD<7:0> is the multiplexed address/data bus and A<7:0> is the latched address output bus. When this signal is low, AD<7:0> is the data bus and A<7:0> is the address input bus. An internal pullup allows this signal to be legally "floated" to select the default multiplexed bus mode of operation.
SH/L*		I	STROBE HIGH ACTIVE/LOW ACTIVE: This input selects the polarity of the DS (Data Strobe) and AS (Address Strobe) input if in Motorola mode. A high on the input will select high active DS and AS while a low on the input will select low active DS and AS. An internal pullup allows this signal to be legally "floated" to select the default high active mode of operation.
CS*		I	CHIP SELECT: This signal must be asserted for all microcontroller accesses to the CL-SH4400's registers.
ALE/AS		I	ADDRESS LATCH ENABLE/ADDRESS STROBE: On the trailing edge of this signal, the CL-SH4400 latches the address present on the AD or A bus, as selected by M/NM*.

SYMBOL	PIN	TYPE	DESCRIPTION
WR*/R-W*		I	WRITE STROBE/READ_WRITE: When I/MC* is high, this signal is WR*; when CS* and WR* are asserted, the data on the AD bus will be written to the specified register. When I/MC* is low, this signal is R-W*, which determines the direction of data transfer when accessing the CL-SH4400's registers. When CS* and DS are asserted and R-W* is high, a register read operation is in progress. When CS* and DS are asserted and R-W* is low, a register write operation is in progress.
RD*/DS		I	READ STROBE/DATA STROBE: When I/MC* is high, this signal is RD*; when CS* and RD* are asserted, the data from the specified register will be driven onto the AD bus. When I/MC* is low, this signal is DS, which determines the data timing of a register access. When CS* is asserted and R-W* is high, the leading edge of DS indicates when the CL-SH4400 may start driving data onto the AD bus. When CS* is asserted and R-W* is low, the trailing edge of DS indicates when the CL-SH4400 may latch data from the AD bus.
RDY/DTACK*		OD	READY/DATA ACKNOWLEDGE: When I/MC* is high, this signal is the microcontroller Ready line; when this signal is deasserted, the local microcontroller shall insert wait states to allow time for the CL-SH4400 to complete the requested access. When I/MC* is low, this signal is the Data Acknowledge signal; this signal is asserted when the CL-SH4400 has completed the requested access.
AD<7:0>		I/O	MICROCONTROLLER ADDRESS/DATA BUS: When M/NM* is high, AD<7:0> is the multiplexed address/data bus. When M/NM* is low, AD<7:0> is the data bus.
A<7:0>		I/O	MICROCONTROLLER ADDRESS: When M/NM* is high, A<7:0> is the latched address output bus; it presents the eight address bits latched by ALE. These signals are provided for general system use and are available regardless of the state of CS*. When M/NM* is low, A<7:0> is the address input bus.

2.2 VM6400 Interface Pins

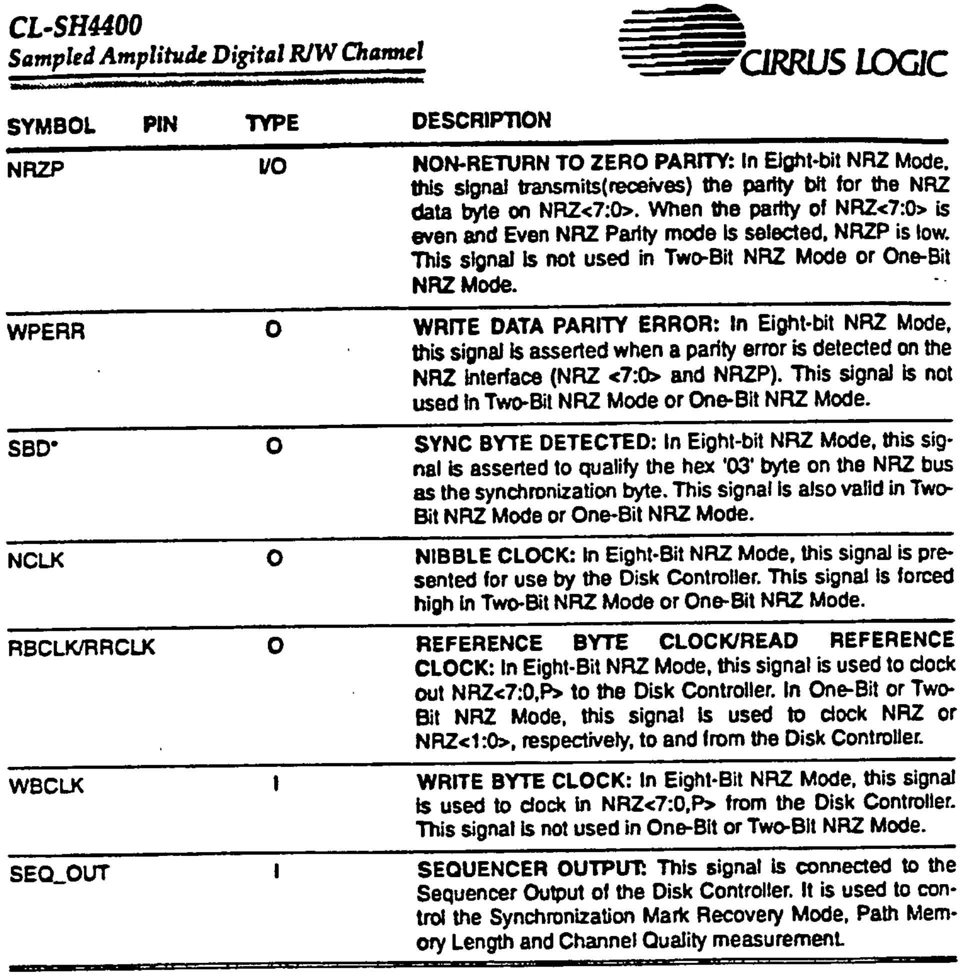
SYMBOL	PIN	TYPE	DESCRIPTION
DRDP<5:0> DRDN<5:0>		I	**DIGITIZED READ DATA:** This differential bus carries the sampled read signal in two's-complement form. These inputs must be driven to a logic '0' when not in use or whenever the sample value is invalid.
RDCLKP RDCLKN		I	**READ CLOCK:** Both edges of this differential signal clocks DRD into the CL-SH4400. RD_CLK is one-half the channel bit rate. This clock must be glitch free and driven to a valid state when not in use.
F3CLK*		I	**F/3 CLOCK:** This signal is used in generating RBCLK/RRCLK and NCLK. F3CLK is one-third or two-thirds the channel bit rate. This clock must be glitch free and driven to a valid state when not in use.
WRRCLK*		I	**WRITE REFERENCE CLOCK:** This signal clocks the RLL encoder of the CL-SH4400. WRRCLK is one-half the channel bit rate. This clock must be glitch free and driven to a valid state when not in use.
ENDAT<0>/SER_CLK		O	**ENCODED DATA BIT 0/SERIAL CLOCK:** While SER_ENA is deasserted, this signal carries the least-significant RLL-encoded write data bit to the VM6400 and is writtem to the disk after the most-significant bit. While SER_ENA is asserted, this signal clocks SER_DAT between the CL-SH4400 and the VM6400.
ENDAT<1>/SER_DAT		I/O	**ENCODED DATA BIT 1/SERIAL DATA:** While SER_ENA is deasserted, this signal carries the most-significant RLL-encoded write data bit to the VM6400 and is written to the disk before the least-significant bit. While SER_ENA is asserted, this signal transfers control information between the CL-SH4400 and the VM6400.
SER_ENA		O	**SERIAL ENABLE:** This signal is asserted for serial control interface operations.
FCTL<4:0>		O	**FREQUENCY CONTROL:** This bus controls the VM6400's variable frequency oscillator/timing recovery circuitry.
ACQ		O	**ACQUISITION MODE:** Controls the range and precision of the FCTL bus.
VGAC<4:0>		O	**VARIABLE GAIN AMPLIFIER CONTROL:** This bus controls the fine oain of the VGA in the VM6400.

2.3 Preamp Interface Pins

SYMBOL	PIN	TYPE	DESCRIPTION
P_CS*		O	**PREAMP CHIP SELECT:** This signal is asserted while RG/RG* or WG/WG* is asserted.
RGO*		O	**READ GATE OUT:** This signal is asserted while RG/RG* is asserted.
WGO*		O	**WRITE GATE OUT:** This signal is asserted while WG/WG* is asserted.

2.4 Disk Controller Interface Pins

SYMBOL	PIN	TYPE	DESCRIPTION
RG/RG*		I	**READ GATE:** When this signal is asserted, the read path circuitry is enabled.
WG/WG*		I	**WRITE GATE:** When this signal is asserted, the write path circuitry is enabled.
NRZ<0>/NRZ		I/O	**NON-RETURN TO ZERO BIT 0/NRZ Data:** In Eight-Bit NRZ Mode and Two-Bit NRZ Mode, this signal transmits(receives) the least-significant unencoded read (write) data bit to(from) the Disk Controller. In One-Bit NRZ Mode, this bit serially transmits(receives) the unencoded read(write) data to(from) the Disk Controller. The tri-state driver is enabled by assertion of RG/RG*.
NRZ<1>		I/O	**NON-RETURN TO ZERO BIT 1:** In Eight-Bit and Two-Bit NRZ Modes, this signal transmits(receives) the second-least-significant unencoded read(write) data bit to(from) the Disk Controller. The tri-state driver is enabled by assertion of RG/RG*. This signal is not used in One-Bit NRZ Mode.
NRZ<7:2>		I/O	**NON-RETURN TO ZERO BITS <7:2>:** In Eight-Bit NRZ Mode, these signals transmit(receive) the six most-significant unencoded read(write) data bits to(from) the Disk Controller. These signals are not used in Two-Bit NRZ Mode or One-Bit NRZ Mode.



CL-SH4400
Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

SYMBOL	PIN	TYPE	DESCRIPTION
NRZP		I/O	**NON-RETURN TO ZERO PARITY:** In Eight-bit NRZ Mode, this signal transmits(receives) the parity bit for the NRZ data byte on NRZ<7:0>. When the parity of NRZ<7:0> is even and Even NRZ Parity mode is selected, NRZP is low. This signal is not used in Two-Bit NRZ Mode or One-Bit NRZ Mode.
WPERR		O	**WRITE DATA PARITY ERROR:** In Eight-bit NRZ Mode, this signal is asserted when a parity error is detected on the NRZ interface (NRZ <7:0> and NRZP). This signal is not used in Two-Bit NRZ Mode or One-Bit NRZ Mode.
SBD*		O	**SYNC BYTE DETECTED:** In Eight-bit NRZ Mode, this signal is asserted to qualify the hex '03' byte on the NRZ bus as the synchronization byte. This signal is also valid in Two-Bit NRZ Mode or One-Bit NRZ Mode.
NCLK		O	**NIBBLE CLOCK:** In Eight-Bit NRZ Mode, this signal is presented for use by the Disk Controller. This signal is forced high in Two-Bit NRZ Mode or One-Bit NRZ Mode.
RBCLK/RRCLK		O	**REFERENCE BYTE CLOCK/READ REFERENCE CLOCK:** In Eight-Bit NRZ Mode, this signal is used to clock out NRZ<7:0,P> to the Disk Controller. In One-Bit or Two-Bit NRZ Mode, this signal is used to clock NRZ or NRZ<1:0>, respectively, to and from the Disk Controller.
WBCLK		I	**WRITE BYTE CLOCK:** In Eight-Bit NRZ Mode, this signal is used to clock in NRZ<7:0,P> from the Disk Controller. This signal is not used in One-Bit or Two-Bit NRZ Mode.
SEQ_OUT		I	**SEQUENCER OUTPUT:** This signal is connected to the Sequencer Output of the Disk Controller. It is used to control the Synchronization Mark Recovery Mode, Path Memory Length and Channel Quality measurement.

2.5 Power, Ground, and Miscellaneous Pins

SYMBOL	PIN	TYPE	DESCRIPTION
+5V		N/A	**POWER SUPPLY** (V_{CC}).
V_{SS}		N/A	**GROUND**
N/C		N/A	**NO CONNECT**
ENA		I	**ENABLE:** Deassertion of this signal places the CL-SH4400 in a power-down state.
CLKIN		I	**CLOCK IN:** This signal is used to develop the clock for the VM6400 serial control interface.

3. REGISTER TABLES

<TBD>

4. FUNCTIONAL DESCRIPTION

An overview of the interaction of the CL-SH4400, VM6400, and the Disk Controller during write and read operations is given below, followed by more detailed descriptions of individual functional blocks.

The major functional blocks of the CL-SH4400 include:

- NRZ Data Interface
- Transition Detector
- Gain Control
- Timing Recovery
- Digital Equalizer
- SofTarget™ Sequence Detector
- Synchronization Mark Detector
- RLL Encoder/Decoder
- Pointer Generation
- Channel Quality
- Serial Control Interface

4.1 Write Operation

When Write Gate is asserted, the Disk Controller must first emit zeros on the NRZ Data interface for the desired length of the preamble. The RLL Encoder in the CL-SH4400 creates the preamble pattern specified by the Preamble Select bits ("1010...", "100100...", or "10001000...") and emits it on the Encoded Data interface. The Disk Controller must then emit a number of hex "FC" bytes on the NRZ Data interface equal to the Synchronization Mark Length, during which the CL-SH4400 emits the Synchronization Mark Pattern on the Encoded Data interface. If necessary, the Synchronization Mark is delayed by one or two channel bits to allow the last "100" or "1000" preamble group to be completed. After it has transferred the proper number of synchronization bytes, the Disk controller must emit data bytes on the NRZ Data interface. If the Data Randomizer is enabled, the data bytes are randomized to equalize the probability of occurrence of worst-case patterns. The RLL Encoder in the CL-SH4400 encodes the data bits and the encoded data bits are emitted on the Encoded Data interface. The preamble bits, Synchronization Mark bits, and encoded data bits are all processed through the VM6400's write precompensation circuitry and transmitted to the write head.

4.1.1 Write Precompensation

Due to the nature of the SofTarget™ Sequence Detector, which takes linear intersymbol interference into account when calculating path metrics, write precompensation for linear intersymbol interference is not required. However, demagnetizing effects during write can cause a shift in the location of a transition which is very close to the preceding transition. The VM6400 provides write precompensation to counter this effect.

4.2 Read Operation

During a read operation, the VM6400 passes the signal from the read/write preamplifier through a variable-gain amplifier and a tunable active filter, then emits digital samples from a six-bit flash analog-to-digital converter. The samples are passed to the CL-SH4400 at the rate of one sample every channel bit interval. In the CL-SH4400, the samples are fed to the programmable digital equalizer which further conditions the signal. The sample points of the analog to digital converter are determined by an oscillator internal to the VM6400 whose frequency/phase is controlled by the CL-SH4400 to maintain proper sampling. The digitally-equalized samples are fed to the SofTarget™ Sequence Detector in the CL-SH4400. The output of the SofTarget™ Sequence Detector is used to detect the Synchronization Mark and to feed the RLL Decoder, whose output is the NRZ data which is transmitted to the Disk Controller on the NRZ Data interface. If the data randomizer is enabled, the data is de-randomized to restore the original user data before being sent to the disk controller.

4.2.1 Number Representation and Arithmetic

The two's complement number system is used wherever negative and positive numbers must both be represented. The input signal samples from the analog-to-digital converter are six-bit two's complement numbers interpreted as having a binary point after the second most significant bit: SX.XXXX. Thus the range of representable numbers is from -2 to +31/16 in steps of 1/16. Note that the binary point is "virtual", having no existence in silicon.

Rounding and truncation are performed as considered appropriate in each internal operation on a case-by-case basis. In some cases, intermediate results are carried with greater precision than the final result. Overflow is guarded against by saturating summers, by algorithmic bounds, or by probability of occurrence as appropriate in each internal operation.

4.3 NRZ Data Interface

The Eight-Bit NRZ Mode allows the CL-SH4400 to connect to a disk controller which uses an eight-bit NRZ data interface. When in the Eight-Bit NRZ Mode, the VM6400 provides the F3CLK signal as a 1/3 channel rate clock which the CL-SH4400 divides by two in frequency in developing NCLK and divides by four in developing RBCLK, which the Disk Controller returns as WBCLK.

The Two-bit NRZ Mode allows the CL-SH4400 to connect to a disk controller which uses a two-bit NRZ data interface. When in the Two-Bit NRZ Mode, the VM6400 provides the F3CLK signal as a 1/3 channel rate clock which the CL-SH4400 uses to develop RRCLK.

The One-bit NRZ Mode allows the CL-SH4400 to connect to a disk controller which uses a one-bit serial NRZ data interface. When in the One-Bit NRZ Mode, the VM6400 provides the F3CLK signal as a 2/3 channel rate clock which the CL-SH4400 uses to develop RRCLK and divides by two to developed its internal two-bit NRZ clock. Data rate in this mode is limited to 40Mbits/sec.

4.4 Transition Detector

Gain Control and Timing Recovery both require information about the locations of transition responses in the sample stream. Because the gain and timing loops cannot tolerate the delay of the SofTarget™ Sequence Detector, a simple transition detector is provided. This transition detector is not normally used for data recovery, although its output may be selected in place of the SofTarget™ Sequence Detector output by setting a control register bit. The peak qualification threshold, V, is also stored in a control register. Table 4.4-1 shows the conditions which must be satisfied to detect a transition during sample period n under various conditions, where:

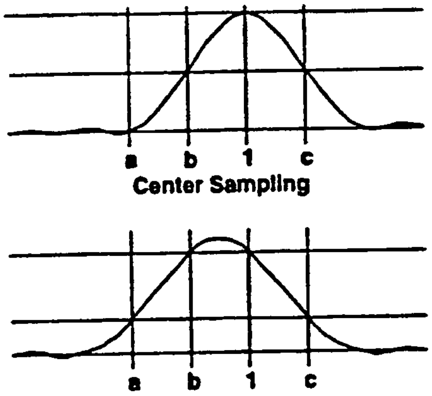
Y_i is the signal sample for period i, and
V is the Qualification Threshold.

For an isolated transition response sample sequence of (a,b,1,c), the transition detector indicates a transition when it receives the fourth sample (c). The choice of either condition (3a) or (3b) during center-samsition pattern. Condition (3b) is intended for use with a '10001000...' acquisition pattern. Either condition might be used for a '100100...' acquisition pattern.

Table 4.4-1 Transition Detection Rules

	Acquisition	Acquisition	Tracking	Tracking
SAMPLING	SGN(Y_{n-1})= 1	SGN(Y_{n-1}) = -1	SGN(Y_{n-1}) = 1	SGN(Y_{n-1}) = -1
Side sampled	1) $Y_{n-2} > Y_n$ 2) $Y_{n-1} > Y_{n-3}$ 3) $Y_{n-2} > Y_{n-4}$	1) $Y_{n-2} < Y_n$ 2) $Y_{n-1} < Y_{n-3}$ 3) $Y_{n-2} < Y_{n-4}$	1) $Y_{n-2} > Y_n$ 2) $Y_{n-1} > Y_{n-3}$ 3) $Y_{n-1} > V$	1) $Y_{n-2} < Y_n$ 2) $Y_{n-1} < Y_{n-3}$ 3) $Y_{n-1} < -V$
Center sampled	1) $Y_{n-1} > Y_n$ 2) $Y_{n-1} > Y_{n-2}$ 3a) $Y_{n-2} > Y_{n-3}$ 3b) $Y_{n-2} > Y_{n-4}$	1) $Y_{n-1} < Y_n$ 2) $Y_{n-1} < Y_{n-2}$ 3a) $Y_{n-2} < Y_{n-3}$ 3b) $Y_{n-2} < Y_{n-4}$	1) $Y_{n-1} > Y_n$ 2) $Y_{n-1} > Y_{n-2}$ 3) $Y_{n-1} > V$	1) $Y_{n-1} < Y_n$ 2) $Y_{n-1} < Y_{n-2}$ 3) $Y_{n-1} < -V$

Figure 4.4-1 illustrates center sampling and side sampling for an isolated pulse.



Center Sampling

Side Sampling

Figure 4.4-1

4.5 Gain Control

The differential variable gain amplifier (VGA) in the VM6400 amplifies the differential input signal from the read preamp to a consistent level which is fed to the tunable active filter. The VGA gain is controlled by a digital to analog converter (DAC) with coarse gain statically controlled by a register in the VM6400 and fine gain dynamically controlled by the Gain Control circuit of the CL-SH4400 through the VGAC bus. The Gain Control circuit is designed for symmetrical attack and decay. The VGAC bus will saturate instead of overflow or underflow.

The Gain Control circuit provides three modes: acquisition, tracking, and hold. In acquisition mode, which is automatically entered on the leading edge of Read Gate, the loop should be programmed to respond quickly in order to allow fast acquisition.

Proper gain is assumed to have been established a programmable number of clocks after the leading edge of Read Gate. The Gain Control circuit then switches to a tracking mode in which the loop should be programmed to respond more slowly in order to minimize gain errors over data due to noise. This change in response is accomplished by switching to another set of loop filter coefficients. It is also possible to freeze the gain during tracking mode by setting the loop gain to zero.

In hold mode, which is entered while Read Gate is deasserted, Gain Control action is suspended, allowing the VGAC bus to "coast" at its current gain level, e.g. over gaps or servo bursts. Gain does not "droop" in hold mode due to the digital nature of the gain control loop.

4.5.1 Gain Detector

The Gain Detector calculates the difference in amplitude between the desired and actual signal levels. A gain update is made whenever a transition is detected. The gain update is negative when the gain is too high and positive when the gain is too low. Table 4.5-1 shows how the gain error is calculated when a transition is detected during sample period n, where

SGN(x) is +1 for $x \geq 0$, -1 for $x < 0$,
Y_i is the signal sample for period i,
g_a is the acquisition Gain Set Point value,
g_t is the tracking Gain Set Point value,
a, b, and c are Detector Levels 1, 2, and 3,
and T_i is 1 when a transition is detected during sample
period i and pulse shape compensation is enabled, 0 otherwise.

The nominal value of the Gain Set Point for side sampling is (1 + b); for center sampling, it is (1). The appropriate value of the Gain Set Point for normal operation is determined during calibration. Varying the Gain Set Point may be a suitable tool for retry strategies. For example, a larger Gain Set Point could be used to hold the gain artificially high, which might tend to compensate for drop-out errors (at some cost in signal-to-noise ratio). Note that the Gain Set Point affects the loop gain and gain margin of the Timing Recovery circuit

Table 4.5-1 Gain Error

	Acquisition	Tracking
Side Sampled	$-SGN(Y_{n-1})(Y_{n-1}+Y_{n-2}) + g_a - (a + c)T_{n-2}$	$-SGN(Y_{n-1})(Y_{n-1}+Y_{n-2}) + g_t - (a + c)T_{n-2}$
Center Sampled	$-SGN(Y_{n-1})Y_{n-1} + g_a - aT_{n-2}$	$-SGN(Y_{n-1})Y_{n-1} + g_t - aT_{n-2}$

4.5.2 Gain Compensation

Compensation in the Gain Control loop is accomplished with a digital filter whose structure may be represented by Figure 4.5-1. The adder is designed to saturate rather than wrap on over/under flow.

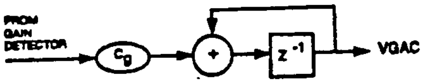


Figure 4.5-1 Gain Control Loop Compensation

The z-domain transfer function of this filter is

$$F_g(z) = \frac{c_g}{z-1}$$

c_g is the loop filter coefficient and z^{-1} represents a delay of two channel bit intervals ($2/f_s$) because this digital filter operates at one-half the channel sample rate. Coefficient c_g is independently programmable for acquisition and tracking.

A linear discrete-time approximation to the open-loop transfer function of the Gain Control loop is

$$G_g(z) = \frac{K_g K_a c_g}{z^n(z-1)}$$

where K_g is the gain of the gain detector (equal to 2 for side sampling, 1 for center sampling), K_a is the control gain of the VGA (equal to 0.46), c_g is the gain of the compensation filter, and n is the number of ($2/f_s$) clock delays in the loop. This model can be used to predict the acquisition settling time, tracking bandwidth, and other properties of the loop for a given filter coefficient c_g.

4.6 Timing Recovery

The Timing Recovery circuit controls the Read Clocks, which are synchronized to the read waveform. The Read Clocks are generated in the VM6400 from a clock produced by a variable frequency oscillator (VFO) which is used to clock an analog-to-digital converter (ADC). The frequency synthesizer in the VM6400 establishes the center frequency of the VFO. A phase detector circuit in the CL-SH4400 digitally computes the phase error in the ADC sampling instants from the sample values. The sequence of measured phase errors is digitally filtered in the CL-SH4400 to produce a frequency control signal which is fed bac to the VM6400 over the FCTL bus. In the VM6400, the digital value from the FCTL bus is converted via DAC to an analog signal which acts to offset the VFO from its center frequency to establish the correc sampling frequency and phase. The phase detector, filter, VFO, and ADC together form a phase-locked loop. The Timing Recovery circuit is designed so that the FCTL bus will saturate instead of overflow o underflow.

The Timing Recovery circuit provides four modes: lock-to-reference, acquisition, tracking, and hold. In lock-to-reference mode, a digital frequency detector is used to lock the read VFO to the frequency synthesizer clock, ensuring small frequency error when acquisition mode is entered.

In acquisition mode, which is automatically entered on the leading edge of Read Gate, the loop should be programmed to respond quickly in order to allow fast acquisition over the preamble. To further decrease acquisition time, the VM6400 provides a zero-phase start.

Timing frequency and phase are assumed to have been acquired a programmable number of clocks after the leading edge of Read Gate. The Timing Recovery circuit then switches to a tracking mode in which the loop should be programmed to respond more slowly in order to minimize timing errors over data due to noise. This change in response is accomplished by switching to another set of loop filter coefficients.

In hold mode, which is entered while Read Gate is deasserted, Timing Recovery action is suspended, allowing the FCTL bus to "coast" at its current frequency, e.g. over gaps or servo bursts.

4.6.1 Phase Detector

The Phase Detector calculates the difference in phase between the desired and actual sampling instants. A timing update is made whenever a transition is detected. The timing update value is positive when the ADC is strobed late, indicating that the VFO frequency should be increased. The timing update value is negative when the ADC is strobed early, indicating that the VFO frequency should be decreased. Table 4.6-1 shows the equations used to calculate the phase error when a transition is detected during sample period n, where

SGN(x) is +1 for $x \geq 0$, -1 for $x < 0$,
Y_i is the signal sample for period i,
t_a is the acquisition Timing Set Point,
t_t is the tracking Timing Set Point,
a, b, and c are Detector Levels 1, 2, and 3, and
T_i is 1 when transition is detected during sample
period i and pulse shape compensation is enabled, 0 otherwise.

The Timing Set Point is used to accommodate pulse asymmetry. Table 4.6-2 shows the nominal values of t_a and t_t. The appropriate values of the Timing Set Point for normal operation are determined during calibration. Varying the values of the Timing Set Point may be a suitable tool for retry strategies.

Table 4.6-1 Phase Error

	Acquisition	Tracking
Side Sampled	$SGN(Y_{n-1})(Y_{n-2} - Y_{n-1}) + t_a$	$SGN(Y_{n-1})(Y_{n-2} - Y_{n-1}) + t_t + cT_{n-2} - aT_{n+2}$
Center Sampled	$SGN(Y_{n-1})(Y_{n-2} - Y_n) + t_a$	$SGN(Y_{n-1})(Y_{n-2} - Y_n) + t_t + cT_{n-2} - bT_{n+2} - aT_{n-3}$

Table 4.6-2 Nominal Timing Set Point

	Acquisition Pattern	Nominal t_a	Nominal t_t
Side Sampled	1010..... 100100...... 10001000....	1 - b + c - a 1 - b 1 - b	1 - b
Center Sampled	1010..... 100100...... 10001000....	2 (c - b) c - b - a c - b	c - b

4.6.2 Timing Compensation

Compensation in the Timing Recovery loop is accomplished with a digital filter whose structure may be represented by Figure 4.6-1.

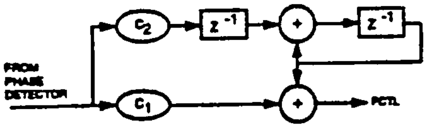


Figure 4.6-1 Timing Recovery Loop Compensation

The z-domain transfer function of this filter is

$$F_t(z) = \frac{c_1 z(z-1) + c_2}{z(z-1)}$$

where z^{-1} represents a delay of two channel bit intervals ($2/f_S$) because this digital filter operates at one-half the channel sample rate. Coefficients c_1 and c_2 are independently programmable for acquisition and tracking.

A linear discrete-time approximation to the open-loop transfer function of the Timing Recovery loop is

$$G_t(z) = \frac{K_p K_o (c_1 z^2 - c_1 z + c_2)}{z^{n+1} (z-1)^2}$$

where K_p is the gain of the phase detector (1 to 2 depending on pulse shape), K_o is the control gain of the VFO (selectable in VM6400) and n is the number of ($2/f_S$) clock delays in the loop. This model can be used to predict the acquisition settling time, tracking bandwidth, and other properties of the loop for a given pair of filter coefficients c_1 and c_2. Given detector levels a, b, and c, the gain of the phase detector may be computed as $K_p = 1 + b - 3(a + c)/2$ for side sampling. For center sampling, the gain of the phase detector is $K_p = 2 - 2a/3 - (b + c)/2$.

4.6.3 Lock to Reference

The center frequency of the read VFO in the VM6400 tracks the frequency synthesizer automatically. This ensures that the frequency control range provided by the FCTL bus is approximately centered around the nominal frequency for each zone. However, the automatic center frequency tracking is not precise enough to facilitate fast, reliable lock to a data preamble. To fine tune the initial frequency of the read VFO and ensure small frequency errors at the start of a lock to preamble, the CL-SH4400/VM6400 chip set provides a Lock to Reference (LTR) mode in which the frequency of the read VFO is adjusted to match the frequency synthesizer. LTR mode should be used whenever the VFO frequency is subject to error, including at power on, after periods of idle time, after a data rate change (inter-zone seek), and after any failure to achieve lock as indicated by failure to detect the synchronization mark.

Lock to Reference (LTR) mode is invoked by setting a bit in a control register in CL-SH4400. LTR mode should not be invoked during any read, write, or servo operation. When the lock is completed the control bit must be manually cleared to return to normal operation.

The CL-SH4400 signals the VM6400 to enter LTR mode by asserting the ACQ line while RGO* is deasserted. The VM6400 then generates a sinusoidal signal with frequency equal to 1/4 of the synthesizer frequency, and samples that signal with the 6-bit flash A/D converter. The CL-SH4400 uses the sample sequence to compute the frequency error between the read VFO (which clocks the A/D converter) and the frequency synthesizer. The frequency error signal is fed back through the digital timing loop filter to drive the error to zero. The direct path coefficient in the timing loop filter (c_1) is set to zero for LTR since inputs represent frequency error measurements rather than the usual phase error measurements. A register is provided to program the value of the accumulator path coefficient (c_2) for LTR mode. By suitable choice of this coefficient, frequency lock times of 100-200 channel bit intervals should be achievable.

4.7 Digital Equalizer

The digital equalizer can modify the equalization done in the analog filter of the VM6400 and, along with it, provides support for changing equalization needs from head to head and zone to zone. The digital equalizer is composed of two specialized digital filters, called the Pulse Shaping Filter and the Spectrum Smoothing Filter. The digital equalizer parameters are loaded by the microcontroller at initialization and during seeks. Using the Channel Quality circuit, a procedure is provided whereby the microcontroller can adapt the digital equalizer parameters. The intent of this feature is to allow the microcontroller to find the best parameter sets (those which produce the minimum error rate) for each disk drive, head, and zone. Since parameter optimization is a slow process, it is done only during manufacturing test and during idle times.

Figure 4.7-1 shows the structure of the Pulse Shaping Filter. The two coefficients PC1 and PC2 are independently programmable.

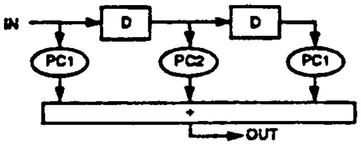


Figure 4.7-1 Pulse Shaping Filter Structure

Figure 4.7-2 shows the Pulse Shaping Filter amplitude response vs. frequency for PC1 = (0, ±1/16, ±1/8, ±1/4) and PC2 = (3/4, 1, 5/4).

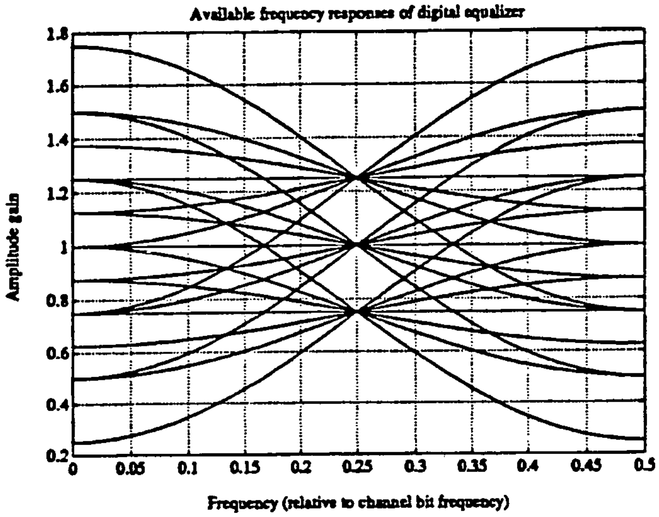


Figure 4.7-2 Pulse Shaping Filter Response

The spectrum smoothing filter is designed to reduce the undershoots from the finite pole tips of a thin-film head, or to reduce the bumps from the secondary gap of a single or double-sided MIG head. Other uses may also be possible. In the frequency domain the filter acts to smooth out the undulation caused by the "Head Bumps", hence the name "Spectrum Smoothing Filter".

Figure 4.7-3 shows the structure of the Spectrum Smoothing Filter. The two delays SD1 and SD2 and the four coefficients SC1, SC2, SC3, and SC4 are independently programmable. The delays may be programmed from 0 to 23 channel bit intervals. The entire Spectrum Smoothing filter or just its precursor-correcting portion (SC1, SC2, and SD1) can be disabled.

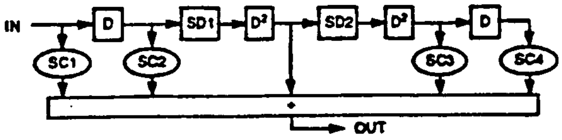


Figure 4.7-3 Spectrum Smoothing Filter Structure

4.8 SofTarget™ Sequence Detector

Partial response is a communications and recording technique used to recover, as much as is practical, the loss associated with intersymbol interference (ISI). Partial response models for magnetic recording channels are typically defined by polynomials of the form

$$(1 - D)(1 + D)^N$$

where D represents a unit delay. The (1 - D) factor represents differentiation by the transducer and the $(1 + D)^N$ factor defines the intersymbol interference. N would be zero for a recording channel without intersymbol interference. N is one for Partial Response Class IV (PR4) and two for Extended Partial Response Class IV (EPR4). Higher orders of (1 + D) are also possible. The expansion of $(1 + D)^N$ defines the sample points of an isolated pulse. Note that when N is even there is a sample at the center of the isolated pulse and that when N is odd all samples are on the sides of the isolated pulse.

Table 4.8-1

N	Partial Response Polynomial	Name	Normalized Sample
0	$(1 - D)$	Dicode	0 1 0
1	$(1 - D)(1 + D)$	PR4	0 1 1 0
2	$(1 - D)(1 + D)^2$	EPR4	0 1/2 1 1/2 0
3	$(1 - D)(1 + D)^3$	EEPR4	0 1/3 1 1 1/3 0
	$(1 - D)(a + bD + D^2 + cD^3)$	SofTarget™	0 a b 1 c 0

We can take a more general view of the polynomial by varying coefficients. For example, replace the side coefficients of [1/2 1 1/2] with K to obtain [K 1 K]. As K varies from 0 to 1/2, the response changes from that associated with the polynomial (1 - D) to that associated with $(1 - D)(1 + D)^2$. The corresponding time and frequency responses are shown in Figure 4.8-1 and Figure 4.8-2, respectively. Similarly, replace the side coefficients of [1/3 1 1 1/3] with K to obtain [K 1 1 K]. As K varies from 0 to 1/3, the response changes from that associated with the polynomial (1 - D)(1 + D) to that associated with $(1 - D)(1 + D)^3$. The corresponding time and frequency responses are shown in Figure 4.8-3 and Figure 4.8-4, respectively. We need not equalize strictly to a target defined by $(1 - D)(1 + D)^N$; a pulse shape in between may be a better match to a particular channel. Additional variability of the coefficients can accommodate asymmetric pulses, rendering phase equalization unnecessary.

At significant user densities, the signal-to-noise ratio required for SofTargef™ Sequence Detector to produce a given error rate is generally several dB lower than that required for digital peak detection. Figure 4.8-5 shows representative bit error rates achieved by the Cirrus Logic SofTarget™ detector programmed for EEPR4 and a digital peak detector as a function of signal-to-noise ratio. Also shown is the performance of each detector with and without the Spectrum Smoothing Filter (SSF).

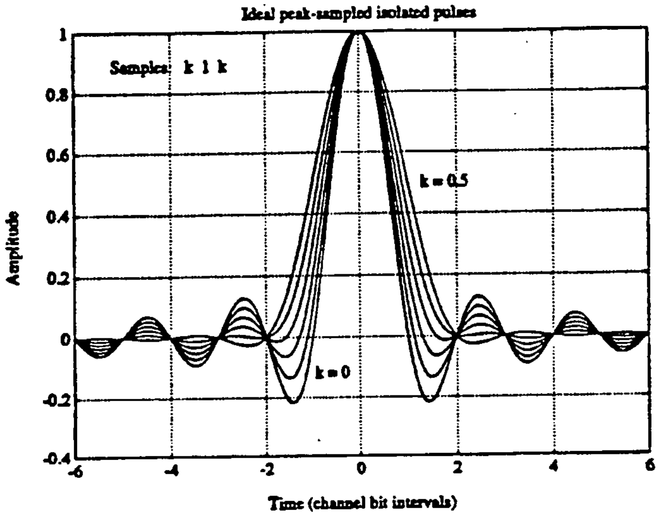


Figure 4.8-1

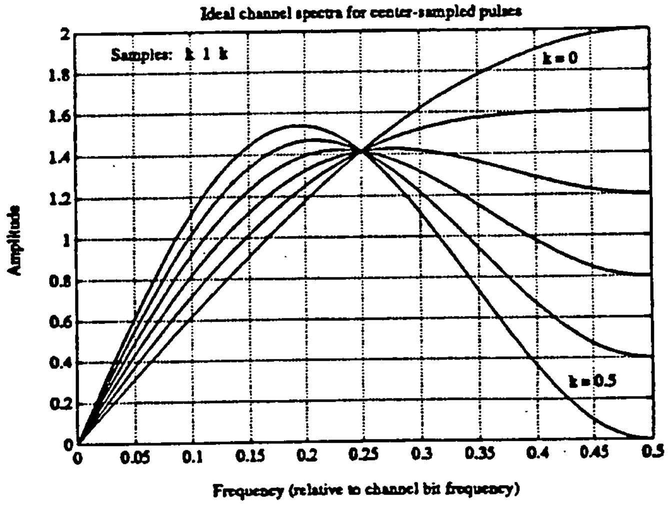


Figure 4.8-2

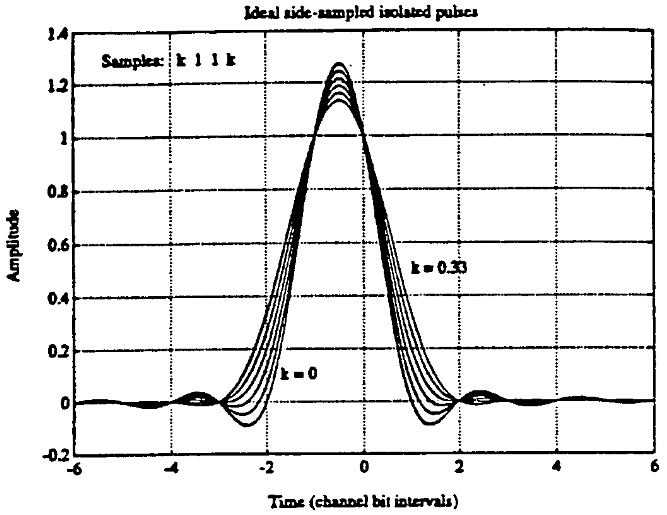


Figure 4.8-3

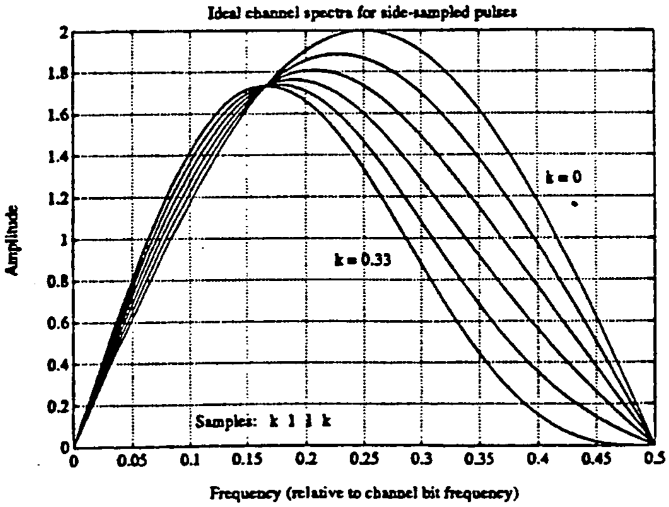


Figure 4.8-4

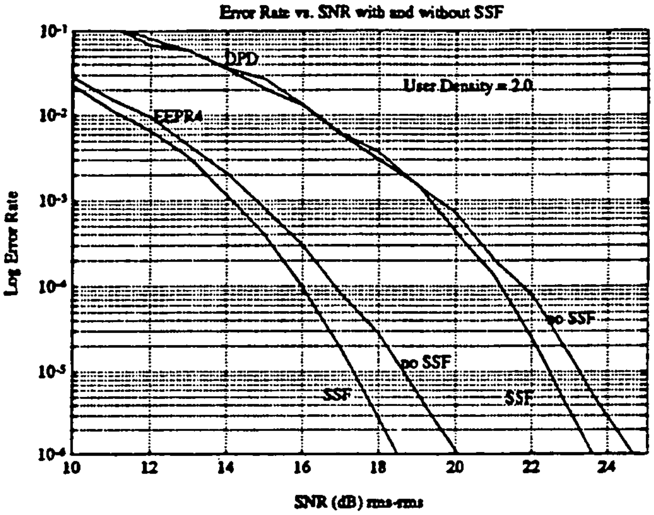


Figure 4.8-5

Figure 4.8-6 shows the SofTarget™ Sequence Detector, which reconstructs the channel-bit stream from the analog- and digitally-equalized samples not on a bit-by-bit basis from each sample in turn but rather with regard to both the current sample and the surrounding sequence of samples according to a decision rule which produces near maximum likelihood detection performance. The detection algorithm determines the best sequence of RLL 1's and 0's corresponding to the pattern of samples.

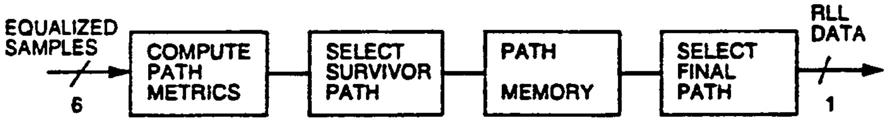


Figure 4.8-6 SofTarget™ Sequence Detector

The SofTarget™ Sequence Detector can be programmed to operate on any channel response which can be well represented by sequences of the form [a b 1 c]. Center-sampled pulses nominally have a < b = c < 1; side-sampled pulses nominally have a < 1, b = 1, and c < 1. Pulse asymmetry is accommodated by specifying a ≠ 0 and b ≠ c for center-sampled pulses, a ≠ c and/or b ≠ 1 for side-sampled pulses. Using the Channel Quality circuit, a procedure is provided whereby the microcontroller may adapt the detector sample levels. The intent of this feature is to allow the microcontroller to find the best detector sample level sets (those which produce the minimum error rate) for each disk drive, head, and zone. Since parameter optimization is a slow process, it is done only during manufacturing test and during idle

The state machine model for the SofTarget™ partial response detector is shown in Figure 4.8-7. Note that the model embodies several kinds of information. First, the isolated pulse sample values a, b, 1, and c are included. Second, the alternating polarity of pulses is enforced. Third, the minimum run-length constraint of d=1 is enforced.

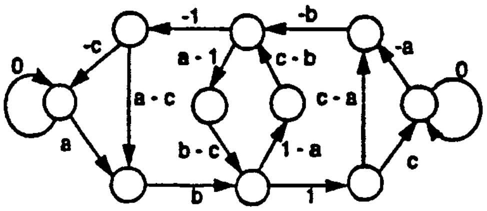


Figure 4.8-7

To program the SofTarget™ detector for PR4, set a = 0, b = 1, and c = 0. For EPR4 set a = 0, b = 0.5, and c = 0.5. For EEPR4 one would set a = 1/3, b = 1, and c = 1/3, although the implementation requires that 1/3 be approximated by 5/16 or 6/16.

4.9 Synchronization Mark Detector

The error-tolerant synchronization strategy implemented in the CL-SH4400 is designed to achieve a level of error tolerance for the synchronization function equal to that achieved for the data field by the combination of the SofTarget™ Sequence Detector in the CL-SH4400 and the error-correction code implemented in the Disk Controller. This strategy employs an error-tolerant Synchronization Mark pattern which is selected for minimum cross-correlation with the preamble and for minimum auto-correlation. The length of the error-tolerant Synchronization Mark and the number of four-channel-bit groups which must be detected are programmable.

When Read Gate is asserted, the CL-SH4400 begins to emit all ones on the NRZ Data interface. When the CL-SH4400 enters tracking mode as specified by the Acquisition Length, it begins to attempt to detect the Synchronization Mark Threshold. When the Synchronization Mark Threshold is met, the CL-SH4400 asserts Sync Byte Detected and emits a single hex "FC" byte over the NRZ Data interface, followed by decoded data bits. Windowing of the detection of the Synchronization Mark is performed in the Disk Controller using the known length of the preamble and Synchronization Mark.

The Synchronization Mark Recovery procedure may be used to recover data when a severe defect has destroyed the entire Synchronization Mark. To use this mode, the Disk Controller asserts the Sequencer Output signal before Read Gate. The CL-SH4400 first goes through a normal gain and timing acquisition procedure while counting channel bits. The Synchronization Mark is assumed to have been detected when the count matches the Synchronization Mark Recovery Count. By varying the Synchronization Mark Recovery Count, the microcontroller can vary the assumed starting point of a header or data area until the correct starting point is tried, whereupon the sector will be recovered if there is no other error beyond the capability of the error correction code in the Disk Controller. A conservative strategy to limit the probability of miscorrection would be to limit error correction to a single burst when the Synchronization Mark Recovery procedure is used.

4.9.1 Synchronization Mark Pattern

The RLL-bit pattern of the error-tolerant Synchronization Mark is programmed by the user in the Synchronization Mark Pattern registers. Cirrus Logic recommends the patterns shown in Table 4.9-1.

Table 4.9-1 Recommended Synchronization Mark Patterns

Acquisition pattern	Synchronization Mark Pattern 47 0
1010.....	10010010 10001001 00010001 01010010 10000100 10101010
100100......	TBD
10001000...	TBD

When the Synchronization Mark Length is programmed for one, two, or three NRZ bytes, the first twelve, twenty-four, or thirty-six bits, respectively, of the pattern are used. When the Synchronization Mark Length is programmed for one, two, three, or four bytes, the Synchronization Mark Threshold should be programmed to require error-free detection of three, four, six, or eight four-channel-bit groups, respectively, in normal operation.

4.10 RLL Encoder/Decoder

During Write operations, NRZ data bits are input from the Disk Controller and encoded according to the rules of a rate 2/3 RLL (1,7) code. The encoded RLL channel bits are then transmitted two at a time to the VM6400 for write precompensation and transmission to the write head.

During Read operations, channel bits are output from the SofTarget™ Sequence Detector and decoded according to the rules of the RLL (1,7) code. The decoded NRZ bits are then output to the Disk Controller.

The RLL (1,7) code is encoded using one data-word look-ahead and one channel-bit look-back, and is decoded using one channel-word look-ahead. Table 4.10-1 shows the encode/decode mapping; x means the complement of the preceding channel bit. The maximum length of an NRZ error burst caused by a single RLL drop-out, drop-in, or bit-shift error is five NRZ bits.

Table 4.10-1 RLL (1,7) Encode/Decode Mapping

NRZ	RLL
00	010
10	x01
11	x00
0100	010001
0101	010000
0110	x00001
0111	x00000

4.10.1 Data Randomizer

The RLL Encoder/Decoder also includes a Data Randomizer which processes unencoded user data to ensure that channel bit patterns with worst-case pattern sensitivity occur no more frequently than would be expected from random user data. The Data Randomizer employs two linear feedback shift registers (LFSR's). One generates a 63-bit sequence which is EXCLUSIVE-OR-ed against the MSB of each pair of data bits. The other generates a 127-bit sequence which is EXCLUSIVE-OR-ed against the LSB of each pair of data bits. The combination of the two LFSR's generates a 8001-bit sequence. The Data Randomizer does not affect error propagation. When the Data Randomizer is enabled, the probability of encountering any specific pattern of length *n* channel bits at a randomly selected location within encoded data is approximately $1/2^n$.

During both read and write operations Sequencer Output must be deasserted over user data and asserted over headers.

4.11 Pointer Generation

Normally, NRZ data bits are transmitted over the NRZ bus. When erasure pointer generation is activated, erasure pointer bits are transmitted. Erasure pointer bits are generated by the detection of a maximum run-length constraint violation in the RLL Decoder. An erasure pointer bit in a given byte could indicate the existence of an error in that byte, a previous byte, or a following byte. When the Channel Quality Mode bit is set and the Channel Quality Input source bits are "0xx", threshold pointer bits are transmitted. Threshold pointer bits are generated by the detection an input sample which differs from its nominal value by more than the Channel Quality Threshold.

4.12 Channel Quality

The nature and quality of the channel comprising the head/disk interface, preamp, tunable active filter, ADC, and digital equalizer can be measured with the Channel Quality circuit, which is provided so that the variable channel parameters (e.g. analog and digital equalization parameters, gain and timing set points, gain control and timing recovery coefficients, and detector levels) may be adjusted to provide the lowest possible error rate. The Channel Quality circuit is a very powerful and flexible measurement tool, providing numerous options to facilitate different kinds of measurements of the channel. By selecting appropriate options it is possible to measure mean squared error (MSE), determine equalized pulse shape, characterize asymmetries between positive and negative pulses, perform margin testing, scan the media for defects, measure DC offset, and measure the performance of the Gain Control and Timing Recovery loops and the zero phase restart. Many of these measurements require that the sequence that was written to the medium be known as the sequence is read. This is accomplished by blocking data input to the CL-SH4400 during a Channel Quality write and generating the written sequence entirely within the Data Randomizer. Then on Channel Quality read, the same sequence can be generated and used in controlling the Channel Quality circuit. Channel Quality measurements may be made using either a pseudo-random data sequence or a programmable two-byte repeating pattern by leaving feedback within the Data Randomizer enabled or disabled, respectively.

For detailed information on the use of the Channel Quality circuit in its various modes, consult CL-SH4400 Application Note xx.

4.13 Path Delays

4.13.1 Write Path Delay

The path delay for the Beacon during a write is given in Table 4.13-1

Table 4.13-1

From	To	Delay In CRRCLK's
NRZ<1:0>	ENDAT<1:0>	TBD

4.13.2 Read Path Delay

The path delays for the Beacon during a read are given in Table 4.13-1 with parameters of Path Memory Length (PML), Spectrum Smoothing Delay 1 (SD1) and Pulse Shaping Filter (PSF).

Table 4.13-1

From	To	Parameters			Delay In RDCLK's
		PML	SD1	PSF	
DRD<5:0>	FCTL<4:0>				TBD
DRD<5:0>	VGAC<4:0>				TBD
DRD<5:0>	NRZ<1:0>	6	Disabled	Enabled	TBD
DRD<5:0>	NRZ<1:0>	12	Disabled	Enabled	TBD
DRD<5:0>	NRZ<1:0>	12	2	Enabled	TBD
DRD<5:0>	NRZ<1:0>	12	2	Disabled	TBD

4.14 Control Interface

When the microcontroller accesses any of the VM6400 Registers, the CL-SH4400 asserts SER_ENA and the register address and data bits are transferred between the CL-SH4400 and the VM6400 over the ENDAT bus. Refer to the Electrical Specification for details.

4.15 Power Management

Deassertion of the Enable input disables all data and clock input receivers. Writing '0' to the Power Management Mode bit disables all non-microcontroller data and clock receivers. Deassertion of the Read gate input disables the RDCLK input receiver, thereby disabling all read circuitry. Deassertion of the Write Gate Input disables the WRCLK input receiver, thereby disabling all write circuitry.

5. OPERATIONAL DESCRIPTION

5.1 Calibration

The sampled-amplitude channel technology employed in the CL-SH4400/VM6400 chip-set makes it possible to implement highly calibrated channels. These channels are calibrated in manufacturing and during idle times and therefore accommodate greater variation in head/media parameters.

In a CL-SH4400/VM6400 implementation, many equalization and detection parameters are under user control. These include:

In VM6400

- Tunable Filter Cutoff
- Tunable Filter Boost
- Zero Phase Start Delay
- Write Precompensation Delay
- VGA Coarse Gain

In CL-SH4400

- Gain Set Point
- Timing Set Point
- Gain Control Loop Coefficients
- Timing Recovery Loop Coefficients
- Digital Equalizer Parameters
- Center/Side Sampling
- Detector Levels
- Transition Detector Threshold

There are many possible tradeoffs; the user decides which parameters are fixed during development, which are to be calibrated during manufacturing, which are to be re-calibrated during idle times, and which are to be varied during retries. Note that a different set of parameters, including the choice of center sampling vs. side sampling, may be used for each head and zone on a given drive. Note also that different sets of parameters may be used within a zone; zones defined by changes in data rate need not correspond with zones defined by parameter changes.

6. REGISTER DESCRIPTIONS

6.1 00H Control/Status Register

Bit 7 **Chip Reset:** Assertion of this bit places the CL-SH4400 in a reset state, deasserts all control outputs and tri-states all bi-directional signals. The CL-SH4400 should be programmed before this bit is written with zero.

Bit 6 **Power Management Mode:**
0: Power-down Mode
1: Operational Mode

Bit 5 **Reserved**

Bit 4 **Clock In to Serial Clock Ratio:** These bits specify the divider used to generate Serial Clock from Clock In.
0: One
1: Two

Bit 3 **Ready Enable (Write)/Serial Interface Busy (Read):**

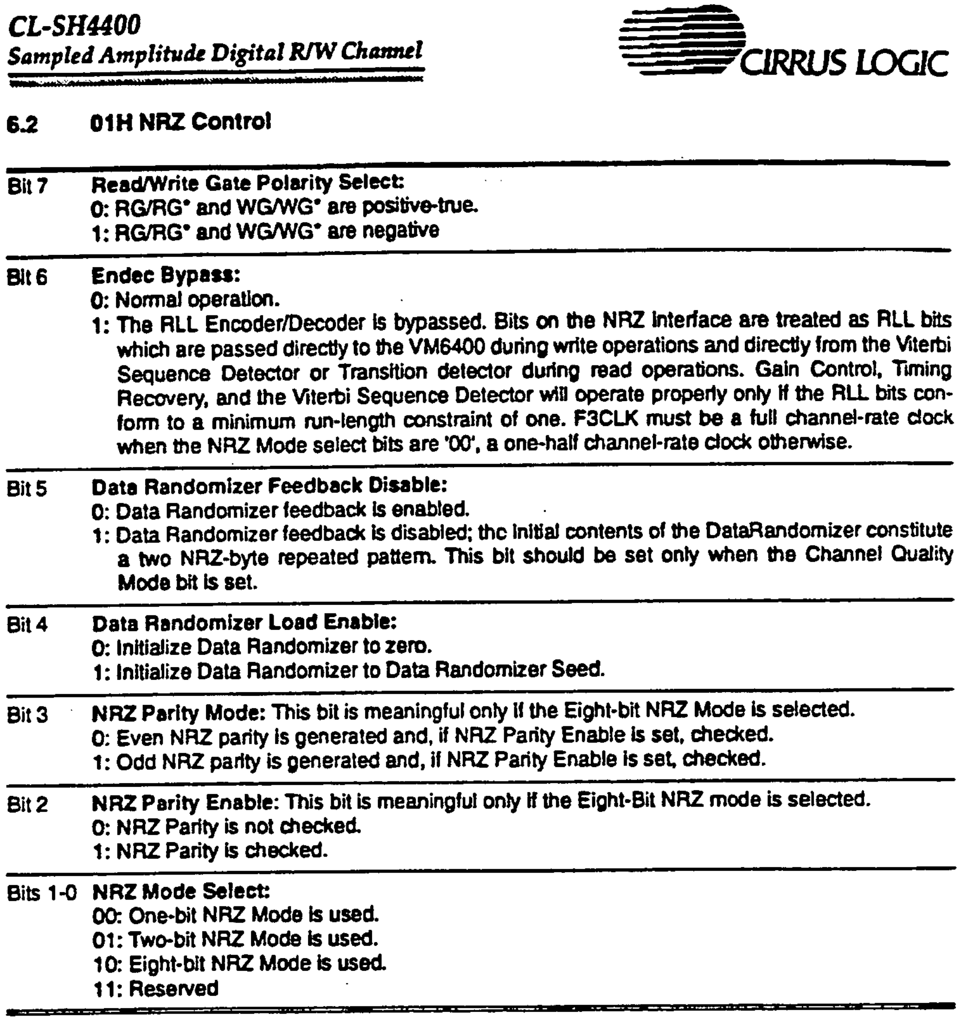
0: When written with a zero Ready is Disabled. This bit is read as '0' when a Serial Control Interface access is allowed. After a Serial Control Interface operation is initiated, this bit is read as '1' until the operation is completed. After a Serial Control Interface read operation is completed, the microcontroller obtains the data bits transferred from the VM6400 by reading any VM6400 Mapped Register, which will initiate another Serial Control Interface read operation.

1: When written with a one Ready is Enabled. This bit is read as '1'. The CL-SH4400 deasserts RDY/DTACK* during Serial Control Interface accesses of VM6400 registers. The deassertion of RDY/DTACK* during a Serial Control Interface write operation forces the microcontroller to wait until the data bits have been transferred to the VM6400. The deassertion of RDY/DTACK* during a Serial Control Interface read operation forces the microcontroller to wait until the data bits have been transferred from the VM6400 and presented on the microcontroller data bus.

Bit 2 **Gain Control Register Saturation (Read only):** This bit is set if the Gain Control register sat urates. It is reset when the control status register is written.

Bit 1 **Acquisition Timing Recovery Register Saturation (Read only):** This bit is set if the Acquisition Timing Recovery register saturates. It is reset when the control status register is written.

Bit 0 **Tracking Timing Recovery Register Saturation (Read only):** This bit is set if the Tracking Timing Recovery register saturates. It is reset when the control status register is written.



6.2 01H NRZ Control

Bit 7 **Read/Write Gate Polarity Select:**
0: RG/RG* and WG/WG* are positive-true.
1: RG/RG* and WG/WG* are negative

Bit 6 **Endec Bypass:**
0: Normal operation.
1: The RLL Encoder/Decoder is bypassed. Bits on the NRZ interface are treated as RLL bits which are passed directly to the VM6400 during write operations and directly from the Viterbi Sequence Detector or Transition detector during read operations. Gain Control, Timing Recovery, and the Viterbi Sequence Detector will operate properly only if the RLL bits conform to a minimum run-length constraint of one. F3CLK must be a full channel-rate clock when the NRZ Mode select bits are '00', a one-half channel-rate clock otherwise.

Bit 5 **Data Randomizer Feedback Disable:**
0: Data Randomizer feedback is enabled.
1: Data Randomizer feedback is disabled; thc initial contents of the DataRandomizer constitute a two NRZ-byte repeated pattem. This bit should be set only when the Channel Quality Mode bit is set.

Bit 4 **Data Randomizer Load Enable:**
0: Initialize Data Randomizer to zero.
1: Initialize Data Randomizer to Data Randomizer Seed.

Bit 3 **NRZ Parity Mode:** This bit is meaningful only if the Eight-bit NRZ Mode is selected.
0: Even NRZ parity is generated and, if NRZ Parity Enable is set, checked.
1: Odd NRZ parity is generated and, if NRZ Parity Enable is set, checked.

Bit 2 **NRZ Parity Enable:** This bit is meaningful only if the Eight-Bit NRZ mode is selected.
0: NRZ Parity is not checked.
1: NRZ Parity is checked.

Bits 1-0 **NRZ Mode Select:**
00: One-bit NRZ Mode is used.
01: Two-bit NRZ Mode is used.
10: Eight-bit NRZ Mode is used.
11: Reserved

6.3 02H Synchronization Mark Control

Bits 7-6 **Preamble Pattern Select**
00: 1010 . . .
01: 100100 . . .
10: 10001000 . . .
11: Reserved

Bits5-4 **Synchronization Mark Length:** These bits specify the length of the Synchronization Mark.
00: One NRZ byte (twelve channel bits).
01: Two NRZ bytes (twenty-four channel bits).
10: Three NRZ bytes (thirty-six channel bits).
11: Four NRZ bytes (forty-eight channel bits).

Bits 3-0 **Synchronization Mark Threshold:** These bits specify one less than the number of four-channel-bit groups in the Synchronization Mark Pattern which must be detected without error in order to detect the Synchronization Mark. Range is 0 to three times the Synchronization Mark Length minus one. Recommended values are 2/3/5/8 for a 1/2/3/4 NRZ-byte Synchronization Mark.

6.4 03H Synchronization Mark Recovery Count

Bit 7 **MUX4 Control:** This bit controls the output of MUX4.
0: Output of Sequence Detector.
1: Output of Transition Detector.

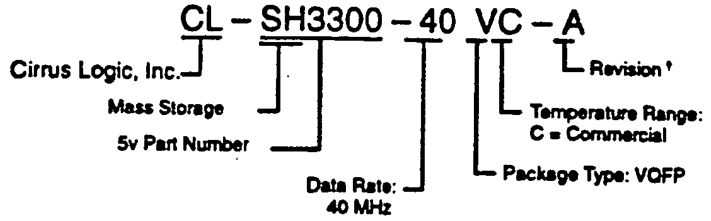
Bits 6-0 **Synchronization Mark Recovery Count:** These bits specify the number of channel-bit periods to wait after the end of acquisition before assuming that the Synchronization Mark has been detected and beginning to decode data. This is done only if Sequencer Out was asserted before the leading edge of Read Gate.

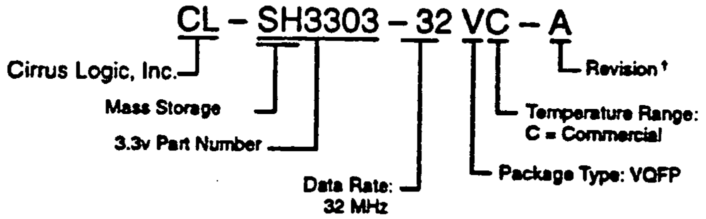
6.5 04H-09H Synchronization Mark Pattern 1-6

Bits 7-0 **Synchronization Mark Pattern Bits:**

Address	Bits	Mark Pattern	Applicable to Synchronization Mark Length
09H	7-0	47-40	4 NRZ bytes
08H	7-4	39-36	4 NRZ bytes
08H	3-0	35-32	3 or 4 NRZ bytes
07H	7-4	31-24	3 or 4 NRZ bytes
06H	7-0	23-16	2, 3, or 4 NRZ bytes
05H	7-4	15-12	2, 3, or 4 NRZ bytes
05H	3-0	11-8	All
04H	7-0	7-0	All

10. ORDERING INFORMATION





Sampled Amplitude Digital R/W Channel

FEATURES

Channel Flexibility

- The CL-SH4400˚provides the digital core of a highly flexible, high-performance sampled-amplitude read/write channel at channel rates up to 96 MHz.

Complete RLL Detector

- Rate 2/3 RLL (1,7) recording code
- Programmable digital equalization
- Programmable SofTarget™ sequence detection
- Digital gain control
- Digital timing recovery

Error Tolerance

- Error-tolerant synchronization
- Channel quality circuitry for error rate testing and filter/detector calibration
- Erasure pointer generation

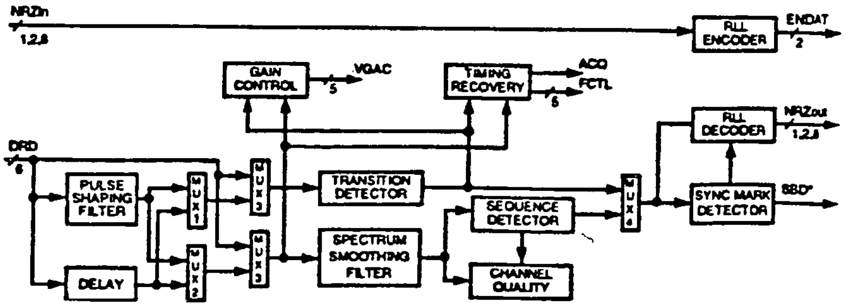
Head Support

| • Monolithic, composite, thin-film, MIG and MR

OVERVIEW

The CL-SH4400 Sampled-Amplitude Digital Channel is a VLSI component designed to work with a companion analog part and a Disk Controller to provide the majority of drive and controller hardware necessary to build a state-of-the-art, high-density, magnetic disk drive. The CL-SH4400 implements the digital portion of a sampled-amplitude read/write channel employing advanced partial response polynomials and SofTarget™ sequence detection technology. It supports data rates up to 64 Mb/s.

Functional Block Diagram



6.10 12H Tracking Gain Set Point

Bits 7-6 **Transition Threshold Bits 3-2**: These bits specify the two most-significant bits of the four-bit Transition Threshold (V) used to qualify transitions during tracking. Range is 0 to +15/16.

Bits 5-0 **Tracking Gain Set Point**: These bits specify the value (g_t) used in computing gain errors during tracking. Range is 0 to +63/16.

6.11 13H Gain Control Coefficient

6.12

Bits 7-4 **Acquisition Gain Control Coefficient**: These bits determine the value of the coefficient (cga) used by the Gain Control compensation loop during acquisition. Those values marked with * are likely to be useful only in side sampled mode. Those values marked with ** are not recommended to be used in side sampled mode.

0000: Reserved	0100: Reserved	1000: 1/16	1100: 1/4
0001: Reserved	0101: Reserved	1001: 3/32	1101: 3/8**
0010: Reserved	0110: 1/32*	1010: 1/8	1110: 1/2**
0011: Reserved	0111: 3/64*	1011: 3/16	1111: Reserved

Bits 3-0 **Tracking Gain Control Coefficient**: These bits determine the value of the coefficient (cgt) used by the Gain Control compensation loop during tracking. Those values marked with * are likely to be useful only in side sampled mode. Those values marked with ** are not recommended to be used in side sampled mode.

0000: 0	0100: 1/64	1000: 1/16	1100: Reserved
0001: 0	0101: 3/128	1001: 3/32**	1101: Reserved
0010: 1/128*	0110: 1/32	1010: 1/8**	1110: Reserved
0011: 3/256*	0111: 3/64	1011: Reserved	1111: Reserved

6.13 14H ACQ Timing Set Point/Coefficient 1

Bits 7-5 **Acquisition Timing Recovery Coefficient**: These bits determine the value of coefficient 1 (c_{1a}) used by the Timing Recovery compensation loop during acquisition.

000: Reserved	100: 1/8
001: Reserved	101: 3/16
010: 1/16	110: 1/4
011: 3/32	111: 3/8

Bits 4-0 **Acquisition Timing Set Point**: These bits specify the value (t_a) used in computing phase errors during acquisition. Range is -1 to +15/16.

6.14 15H TRK Timing Set Point/Coefficient 1

Bits 7-5 **Tracking Timing Recovery Coefficient 1:** These bits determine the value of coefficient 1 (c_{1t}) used by the Timing Recovery compensation loop during tracking.

000: 1/32	100: 1/8
001: 3/64	101: 3/16
010: 1/16	110: 1/4
011: 3/32	111: 3/8

Bit 4 **Sample Phase Mode:**

0: Generate samples for a center-sampled pulse, i.e. $a < b = c < 1$; timing is recovered using adjacent-but-one samples.

1: Generate samples for a side-sampled pulse i.e. $a = c < b = 1$; timing is recovered using adjacent samples.

Bits 3-0 **Tracking Timing Set Point:** These bits specify the value (t_t) used in computing phase errors during tracking. Range is -1/2 to +7/16.

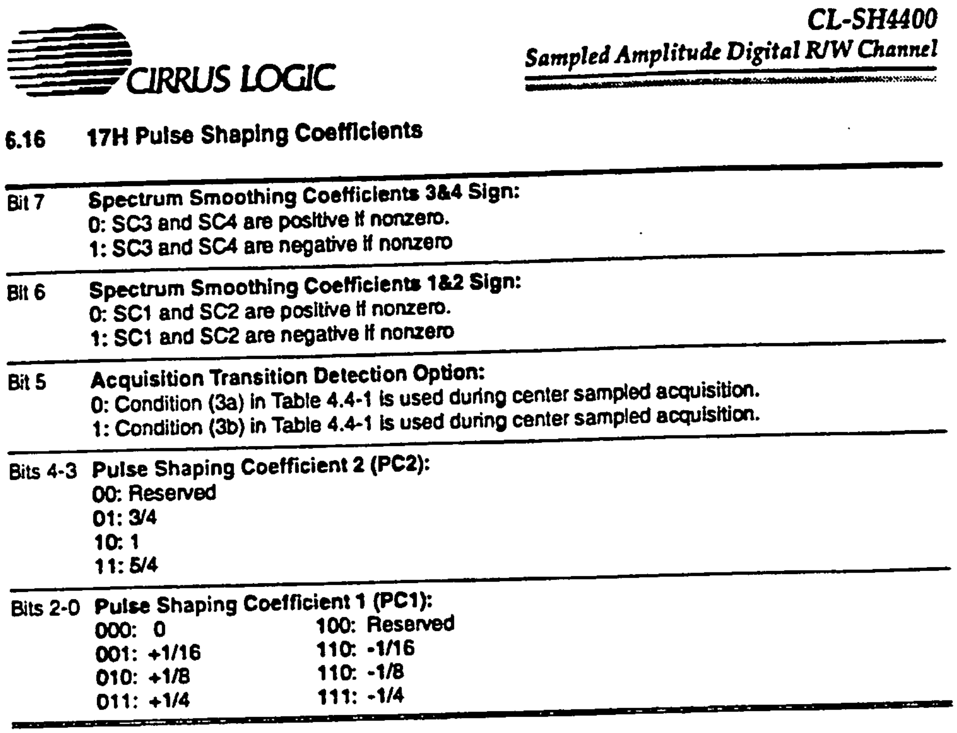
6.15 16H Timing Recovery Coefficient 2

Bits 7-4 **Acquisition Timing Recovery Coefficient 2:** These bits determine the value of coefficient 2 (c_{2a}) used by the Timing Recovery compensation loop during acquisition.

0000: 1/512	0100: 1/128	1000: 1/32	1100: Reserved
0001: 3/1024	0101: 3/256	1001: 3/64	1101: Reserved
0010: 1/256	0110: 1/64	1010: 1/16	1110: Reserved
0011: 3/512	0111: 3/128	1011: 3/32	1111: Reserved

Bits 3-0 **Tracking Timing Recovery Coefficient 2:** These bits determine the value of coefficient 2 (c_{2t}) used by the Timing Recovery compensation loop during tracking.

0000: 1/4096	0100: 1/1024	1000: 1/256	1100: Reserved
0001: 3/8192	0101: 3/2048	1001: 3/512	1101: Reserved
0010: 1/2048	0110: 1/512	1010: 1/128	1110: Reserved
0011: 3/4096	0111: 3/1024	1011: 3/256	1111: Reserved



CIRRUS LOGIC

CL-SH4400

Sampled Amplitude Digital R/W Channel

6.16 17H Pulse Shaping Coefficients

Bit 7	**Spectrum Smoothing Coefficients 3&4 Sign:** 0: SC3 and SC4 are positive if nonzero. 1: SC3 and SC4 are negative if nonzero
Bit 6	**Spectrum Smoothing Coefficients 1&2 Sign:** 0: SC1 and SC2 are positive if nonzero. 1: SC1 and SC2 are negative if nonzero
Bit 5	**Acquisition Transition Detection Option:** 0: Condition (3a) in Table 4.4-1 is used during center sampled acquisition. 1: Condition (3b) in Table 4.4-1 is used during center sampled acquisition.
Bits 4-3	**Pulse Shaping Coefficient 2 (PC2):** 00: Reserved 01: 3/4 10: 1 11: 5/4
Bits 2-0	**Pulse Shaping Coefficient 1 (PC1):** 000: 0 100: Reserved 001: +1/16 110: -1/16 010: +1/8 110: -1/8 011: +1/4 111: -1/4

6.17 18H Spectrum Smoothing Coefficients

When SC1 = SC2 = "00", the precursor-correcting portion of the filter is disabled and the pipeline delay of the Spectrum Smoothing filter is minimized. When SC3 = SC4 = "00", the postcursor-correcting portion of the filter is disabled. The sign of SC1 and SC2 is determined by Spectrum Smoothing Coefficients 1&2 Sign. The sign of SC3 and SC4 is determined by Spectrum Smoothing Coefficients 3&4 Sign.

Bits 7-6	**Spectrum Smoothing Coefficient 4 (SC4):** 00: 0 01: ±1/32 10: ±1/16 11: Reserved
Bits 5-4	**Spectrum Smoothing Coefficient 3 (SC3):** 00: 0 01: ±1/32 10: ±1/16 11: Reserved
Bits 3-2	**Spectrum Smoothing Coefficient 2 (SC2):** 00: 0 01: ±1/32 10: ±1/16 11: Reserved
Bits 1-0	**Spectrum Smoothing Coefficient 1 (SC1):** 00: 0 01: ± 1/32 10: ±1/16 11: Reserved

6.18 19H Spectrum Smoothing Delays

Bits 7-4	**Spectrum Smoothing Delay 2:** Range is 0 to 15. SD2 is equal to the value programmed here plus eight additional bits of delay if the Additional Spectrum Smoothing Delay 2 Enable bit is set.
Bits 3-0	**Spectrum Smoothing Delay 1:** Range is 0 to 15. SD1 is equal to the value programmed here plus eight additional bits of delay if the Additional Spectrum Smoothing Delay 1 Enable bit is set.

6.19 1AH Lock-To-Reference Control

Bit 7 **Lock-To-Reference Enable:**
0: Normal Operation.
1: Lock-To-Reference Mode is enabled.

Bit 6 **Erasure Pointer Enable:**
0: Normal operation; data bits are output on NRZ bus.
1: Erasure pointers are output on NRZ bus.

Bit 5 **Additional Spectrum Smoothing Delay 2 Enable:**
0: Normal Operation.
1: SD2 is increased by eight additional bits of delay.

Bit 4 **Additional Spectrum Smoothing Delay 1 Enable:**
0: Normal Operation.
1: SD1 is increased by eight additional bits of delay.

Bits 3-0 **Lock-To-Reference Coefficient:** These bits determine the value of coefficient 2 (c_{2l}) used by the Timing Recovery compensation loop during lock-to-reference.

0000:	1/512	0100:	1/128	1000:	1/32	1100:	1/8
0001:	3/1024	0101:	3/256	1001:	3/64	1101:	3/16
0010:	1/256	0110:	1/64	1010:	1/16	1110:	1/4
0011:	3/512	0111:	3/128	1011:	3/32	1111:	3/8

6.20 1BH Detector Levels a & c

Bits 7-4 **Detector Level c:** These bits specify the last sample value (c) for a nominal pulse. Supported range is 0 to +12/16.

Bits 3-0 **Detector Level a:** These bits specify the first sample value (a) for a nominal pulse. Supported range is 0 to +8/16.

6.21 1CH Detector Level b/MUX Control Register

Bit 7 **MUX3 Control:** This bit controls the output of MUX3 during tracking; during acquisition, MUX3 selects raw input samples.
0: Raw input samples.
1: Delayed or Pulse Shaping Filtered input samples, as controlled by the MUX1 Control and MUX2 Control bits.

Bit 6 **MUX2 Control:** This bit controls the output of MUX2. Note that the output of MUX2 is not used during acquisition.
0: Pulse Shaping Filtered input samples.
1: Delayed input samples.

Bit 5 **MUX1 Control:** This bit controls the output of MUX1. Note that the output of MUX1 is not used during acquisition.
0: Pulse Shaping Filtered input samples.
1: Delayed input samples.

Bits 4-0 **Detector Level b:** These bits specify the second sample value (b) for a nominal pulse. Supported range is 0 to +24/16.

6.22 1DH Gain Control Loop Register

Bits 7-0 **Gain Control Loop Register:** The value read and written here is the eight most-significant bits of the Gain Control Loop register. When this register is written, the remaining bits of the Gain Control Loop register are cleared.

6.23 1EH ACQ Timing Recovery Loop Register

Bits 7-0 **Acquisition Timing Recovery Loop Register:** The value read and written here is the eight most-significant bits of the Acquisition Timing Loop Recovery Loop register. When this register is written, the remaining bits of the Acquisition Timing Recovery Loop register are cleared.

6.24 1FH TRK Timing Recovery Loop Register

Bits 7-0 **Tracking Timing Recovery Loop Register:** The value read and written here is the eight most-significant bits of the Tracking Timing Recovery Loop register. When this register is written, the remaining bits of the Tracking Timing Recovery Loop register are cleared. All bits of the Tracking Timing Recovery Loop register are automatically cleared at the end of acquisition.

6.25 20H Channel Quality Control 1

Bit 7 **Channel Quality Mode:** When this bit is written with '1', the Channel Quality Count and Channel Quality Output registers are cleared.
0: Normal Operation.
1: Channel Quality Mode. During user data write operations, the NRZ sequence to be encoded and written, after the Synchronization Mark, is generated in the Data Randomizer; NRZ input from the Disk Controller is ignored. During Read Operations, the Data Randomizer must be configured to generate the same sequence.

Bit 6 **Acquisition Quality Enable:**
0: Data Quality Mode. The Channel Quality circuitry is active from detection of the Synchronization Mark to the leading edge of Sequencer Out.
1: Acquisition Quality Mode. The Channel Quality circuitry is active from the leading edge of Read Gate to the end of Acquisition.

Bit 5 **Channel Quality Test Mode:** This bit must be '0' for normal operation.

Bits 4-0 **Channel Quality Threshold:** When the Channel Quality Input Source bits are "0xx", these bits specify the error which must be exceeded in order for a sample to be counted. Range is 0 to +31/16.

6.26 21H Channel Quality Control 2

Bit 7 **Odd Synchronization (Read Only):**
0: The last bit of the previously-detected Synchronization Mark was detected during the second sample of a pair of samples.
1: The last bit of the previously-detected Synchronization Mark was detected during the first sample of a pair of samples.

Bits 6-4 **Channel Quality Input Source:** This bit controls what is counted or accumulated in the Channel Quality Output registers.
000: The events where the received bit does not compare to the expected bit are counted.
001: Samples which are less than their nominal values by more than the Channel Quality Threshold are counted.
010: Samples which are greater than their nominal values by more than the Channel Quality Threshold are counted.
011: Samples which are less than their nominal values by more than the Channel Quality Threshold are counted and samples which are greater than their nominal values by more than the Channel Quality Threshold are counted.
100: Unsquared samples are accumulated.
101: Squared sample errors are accumulated.
110: Squared Timing errors are accumulated.
111: Squared Gain errors are accumulated.

Bits 3-2 **Header Detector Path Memory Length:** These bits specify the length of the path memory in the Viterbi Sequence Detector over headers. Sequencer Output must be deasserted over user data and asserted over headers.
00: Six channel bits
01: Twelve channel bits
10: Eighteen channel bits
11: Twenty-four channel bits

Bits 1-0 **Data Detector Path Memory Length:** These bits specify the length of the path memory in the Viterbi Sequence Detector over user data. Sequencer Output must be deasserted over user data and asserted over headers.
00: Six channel bits
01: Twelve channel bits
10: Eighteen channel bits
11: Twenty-four channel bits

6.27 22H-25H Channel Quality Sample Control 1-4

These registers control which samples contribute to the Channel Quality Count and Channel Quality Output registers. For further information, consult CL-SH4400 Application Note xx.

Sample	Register 22H	Register 23H	Register 24H	Register 25H
0 before +a	40	00	40	40
+a	41	40	40	08
+b	40	18	6A	00
+1	78	00	60	82
+c	60	02	61	04
0 after +c	60	04	40	00
0 before -a	60	00	40	40
-a	61	40	60	08
-b	60	18	4A	00
-1	58	00	40	82
-c	40	02	41	04
0 after-c	40	04	60	00
0 before ±a	00	00	00	40
±a	01	40	00	08
±b	00	18	0A	00
±1	18	00	00	82
±c	00	02	01	04
0 after ±c	00	04	00	00

6.28 26H Channel Quality Count LSB

Bits 7-0 **Channel Quality Count LSB:** This value is the eight least-significant bits of the total number of samples which satisfied the criteria specified by the Channel Quality Sample Control registers since the Channel Quality Mode bit was last written with '1'.

6.29 27H Channel Quality Count MSB

Bits 7-0 **Channel Quality Count MSB:** This value is the eight most-significant bits of the total number of samples which satisfied the criteria specified by the Channel Quality Sample Control registers since the Channel Quality Mode bit was last written with '1'.

6.30 28H Channel Quality Output LSB

Bits 7-0 **Channel Quality Output LSB:** This value is the eight least-significant bits of the Channel Quality Output word counted or accumulated since the Channel Quality Mode bit was last written with '1'.

6.31 29H Channel Quality Output MID

Bits 7-0 **Channel Quality Output MID:** This value is the second eight least-significant bits of the Channel Quality Output word counted or accumulated since the Channel Quality Mode bit was last written with '1'.

6.32 2AH Channel Quality Output MSB

Bits 7-0 **Channel Quality Output MSB:** This value is the eight most-significant bits of the Channel Quality Output word counted or accumulated since the Channel Quality Mode bit was last written with '1'.

6.33 2EH Reserved

6.34 2FH Revision Register

Bits 7-0 **Revision Code (Read only): For Rev. A parts "FFH" will be read.**

6.35 30H-3FH VM6400 Mapped Registers

Bits 7-0 **VM6400 Mapped Register:**

When one of the addresses mapped to the VM6400 is written, a Serial Control Interface write operation is initiated in which the data to be written is latched from the microcontroller data bus and transmitted to the mapped register in the VM6400. If the Ready Enable bit has been written with '1', the CL-SH4400 deasserts RDY/DTACK* until the Serial Control write operation is completed. If the Ready Enable bit has been written with '0', the microcontroller must wait until the Ready Enable bit is read as '0' before initiating another Serial Control Interface operation.

When one of the addresses mapped to the VM6400 is read, a Serial Control Interface read operation is initiated in which the data to be read is transmitted from the mapped register in the VM6400. If the Ready Enable bit has been written with '1', the CL-SH4400 deasserts RDY/DTACK* until the Serial Control read operation for the mapped register is completed and the data bits are presented on the microcontroller data bus. If the Ready Enable bit has been written with '0', the CL-SH4400 presents the data bits transferred for the previous Serial Control Interface read operation on the microcontroller data bus and initiates a Serial Control Interface read operation for the current mapped register. To obtain the data bits transferred for the current mapped register, the microcontroller must wait until the Ready Enable bit is read as '0' and then read the same or another VM6400 Mapped Register.

7. PERFORMANCE SPECIFICATIONS

7.1 Gain Control

Acquisition Time: 48 to 372) $/f_S$ Typical: 48-84

Tracking Bandwidth: 0 to 7.5e-3) f_S

where f_S is the sampling rate.

7.2 Timing Recovery

Acquisition Time: (48 to 372) $/f_S$ Typical: 84-108

Tracking Bandwidth: (1e-3 to 2.5e-2) f_S

where f_S is the sampling rate.

8. INITIALIZATION CONDITIONS

8.1 Registers

When the RST* input is asserted the following register is affected. All other registers remain either undefined or as previously written.

Register	Value
00H	10xx0xxx

8.2 Outputs

When the RST* input is asserted or the Chip Reset bit is set the outputs are deasserted per the following table. If the entry does not have an 'x' entered in the column then that output is unaffected.

Output	RST*	Chip Reset bit
SER_ENA	x	x
ACQ	x	x
P_CS*	x	x
RGO*	x	x
WGO*	x	x
WPERR	x	x
SBD*	x	x

9. ELECTRICAL SPECIFICATIONS

9.1 Absolute Maximum Ratings

Parameter	Minimum	Maximum	Units
Ambient Temperature Under Bias	0	70	°C
Storage Temperature	-65	150	°C
Voltage On Any Pin	-0.5	V_{CC}+0.5	V
Power Dissipation		<TBD>	W
Power Supply Voltage		7.0	V

NOTE: Stresses above those listed may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these conditions, or any conditions outside those indicated in the operational sections of this specification, is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

9.2 DC Characteristics

Symbol	Parameter	Minimum	Maximum	Units
V_{CC}	Supply High Voltage	4.5	5.5	V
V_{ID}	Differential Input Voltage	±0.4		V
V_{IR}	Input Voltage Range-Differential Inputs	$V_{CC}/2$	V_{CC}-0.5	V
V_{IL}	Input Low Voltage	-0.5	0.8	V
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V
V_{OL}	Output Low Voltage @ IOL = 2.0 mA		0.4	V
V_{OH}	Output High Voltage @ IOH = -400 µA	2.4		V
I_{CC}	Supply Current		<TBD>	mA
I_{L}	Input Leakage Current	-10	10	µA
I/O_{L}	I/O Leakage Current	-10	10	µA
C_{IN}	Input Capacitance		10	pF
C_{OUT}	Output Capacitance		10	pF

NOTE: All unused inputs must be tied to GND or V_{CC}.

9.3 AC Characteristics

Unless otherwise specified, the following timings assume that all outputs will drive one Schottky TTL load in parallel with 50 pF and all inputs are at TTL level. The Min. and Max. timings conform to the operating ranges of power supply voltage (5V ±10 percent) and ambient temperature (0 to 70 ° C). All clocks are 60/40 percent maximum/minimum duty cycle. Rise and fall times should not exceed 3 ns.

9.3.1 Reset Assertion Timing Parameters

Symbol	Parameter	Minimum	Maximum	Units
Trpwl	RST* pulse width low	500		ns

9.3.2 Microcontroller Interface Timing Parameters

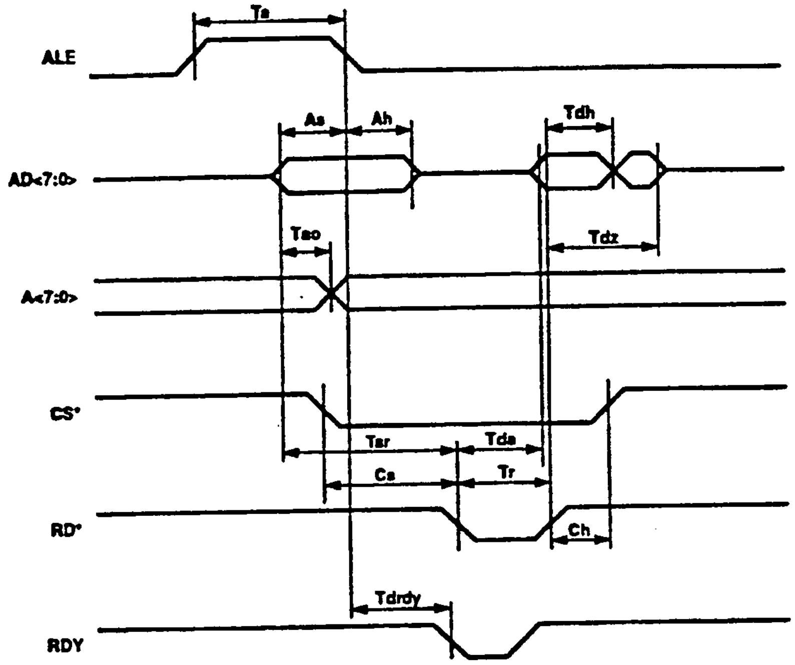
Symbol	Parameter	Minimum	Maximum	Units
Ta	ALE width	15		ns
Taw	Address valid to WR*↓ or DS↑	20		ns
Tar	Address valid to RD*↓ or DS↑	15		ns
Tao	AD<7:0> valid to A<7:0>		20	ns
	(Multiplexed mode)			
Tah	RD* ↑, WR*↑ or DS↓ to A<7:0> invalid	5		ns
	(Nonmultiplexed Mode)			
Tw	WR* width	60		ns
Tr	RD* width	60		ns
As	Address valid to ALE ↓	5		ns
Ah	ALE↓ to address invalid	0		ns
Cs	CS*↓ to RD* ↓, WR* ↓, or DS ↑	0		ns
Ch	RD* ↑, WR*↑ or DS↓ to CS* ↑	5		ns
Wds	Write data valid to WR*↑ or DS ↓	20		ns
Wdh	WR*↑ or DS↓ to Write data invalid	10		ns
Tda	RD*↓ or DS↑ to Read data valid		50	ns
Tdh	RD*↑ or DS↓ to Read data invalid	10		ns
Tdz	RD*↑ or DS↓ to Read data undriven		15	ns
Tds	DS width	60		ns
Tdth	DS↑ to DTACK* ↑		20	ns
Tdrdy	ALE↓ to RDY ↓		30	ns
Tsrw	R_W* valid to DS ↑	20		ns
Thrw	DS↓ to R_W* invalid	20		ns

Note: ↑ indicates rising edge,↓ indicates falling edge.

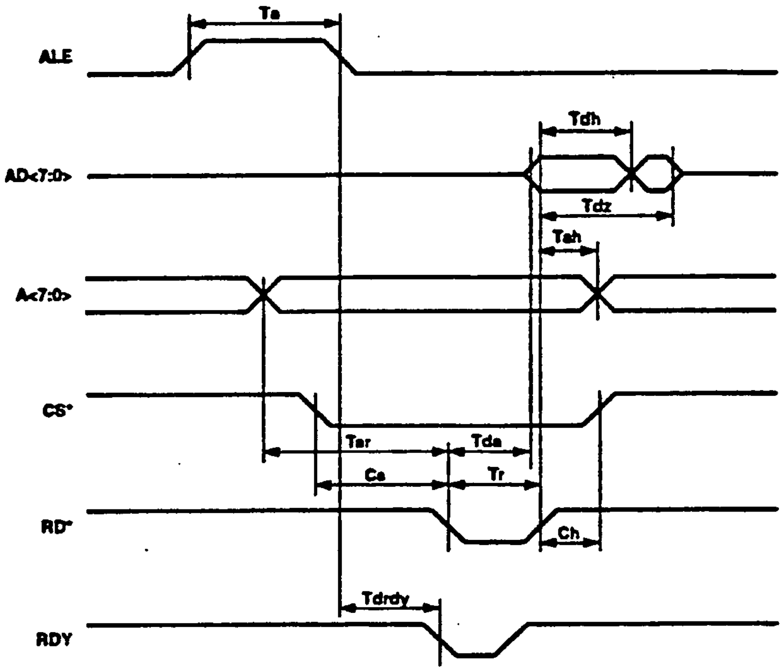
When I/MC* is high, the Intel bus control interface is selected. The timing diagrams in Sections 9.3.2.1 - 9.3.2.4 depict register read and write operations with this interface selected.

When I/MC* is low, the Motorola bus control interface is selected. The timing diagrams in Sections 9.3.2.5 - 9.3.2.8 depict register read and write operations with this interface selected.

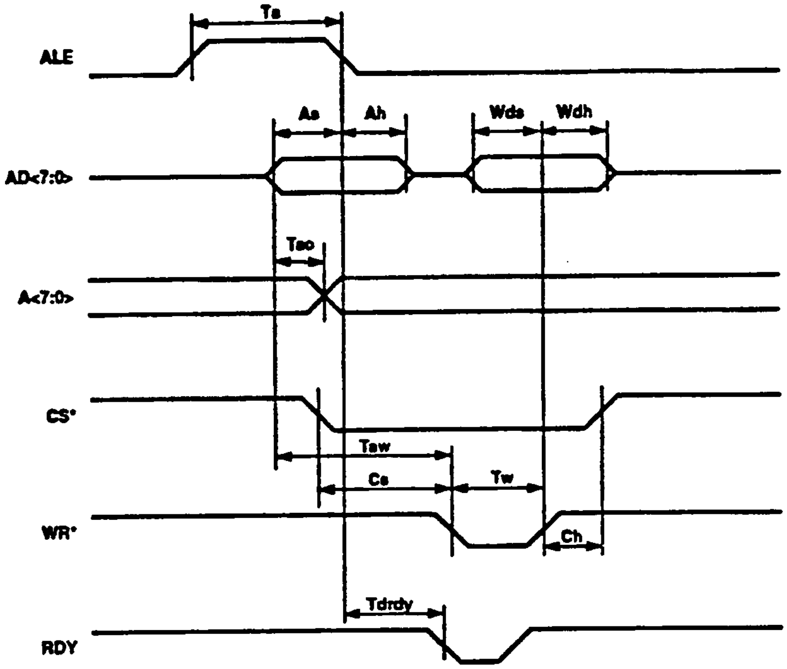
9.3.2.1 Register Read Operation in Intel Multiplexed Mode



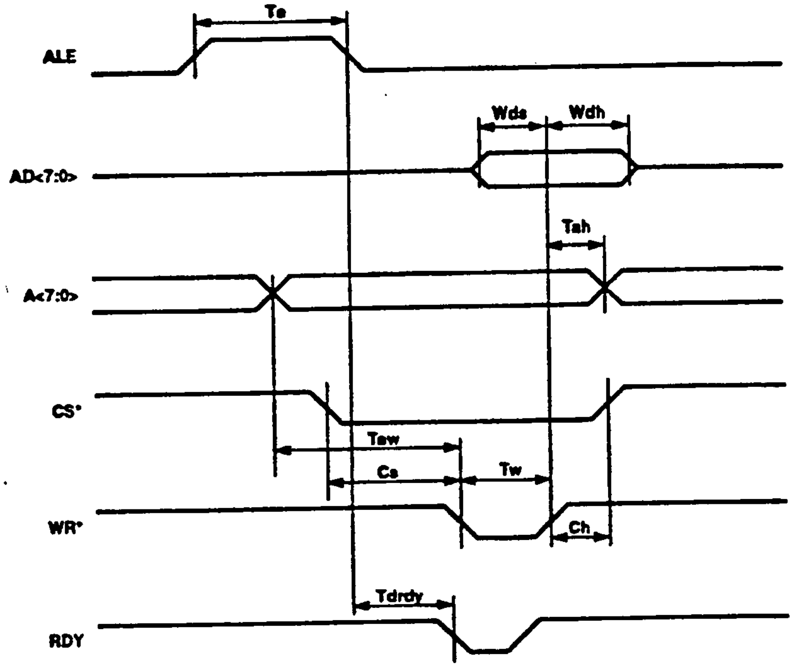
9.3.2.2 Register Read Operation in Intel Nonmultiplexed Mode



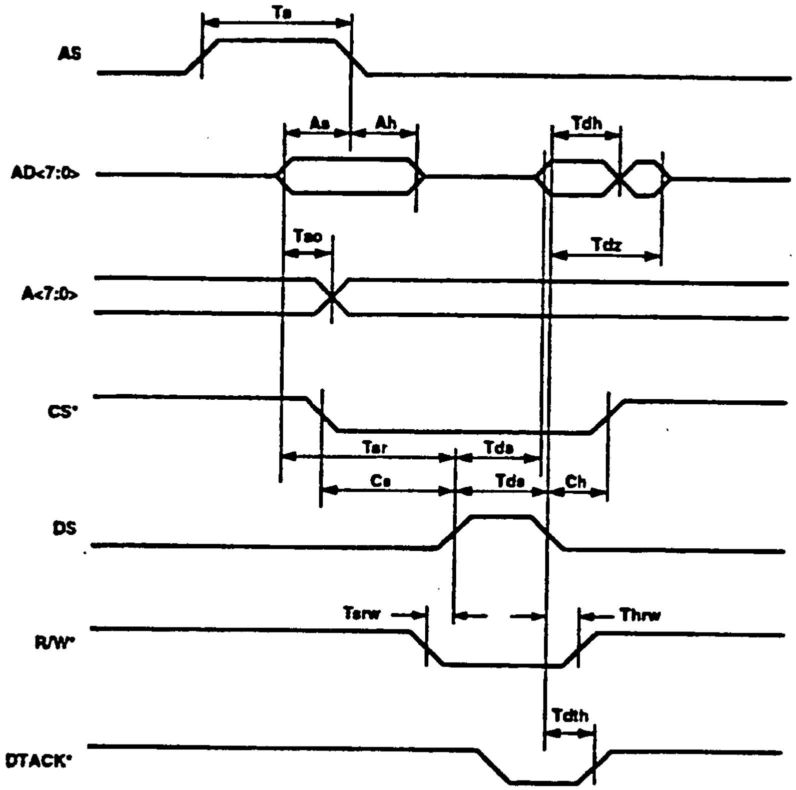
9.3.2.3 Register Write Operation in Intel Multiplexed Mode



9.3.2.4 Register Write Operation in Intel Nonmultiplexed Mode

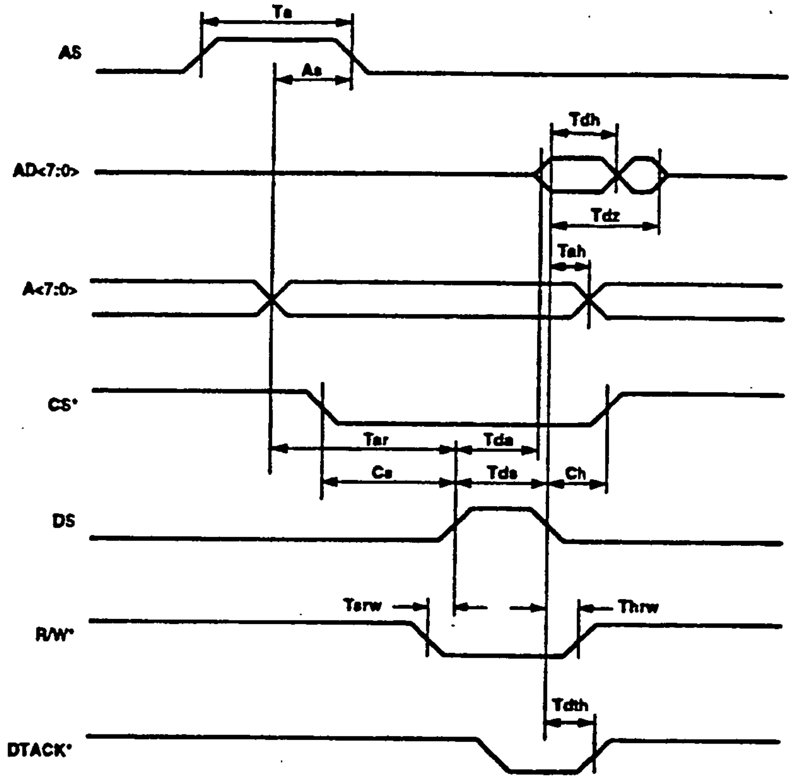


9.3.2.5 Register Read Operation in Motorola Multiplexed Mode



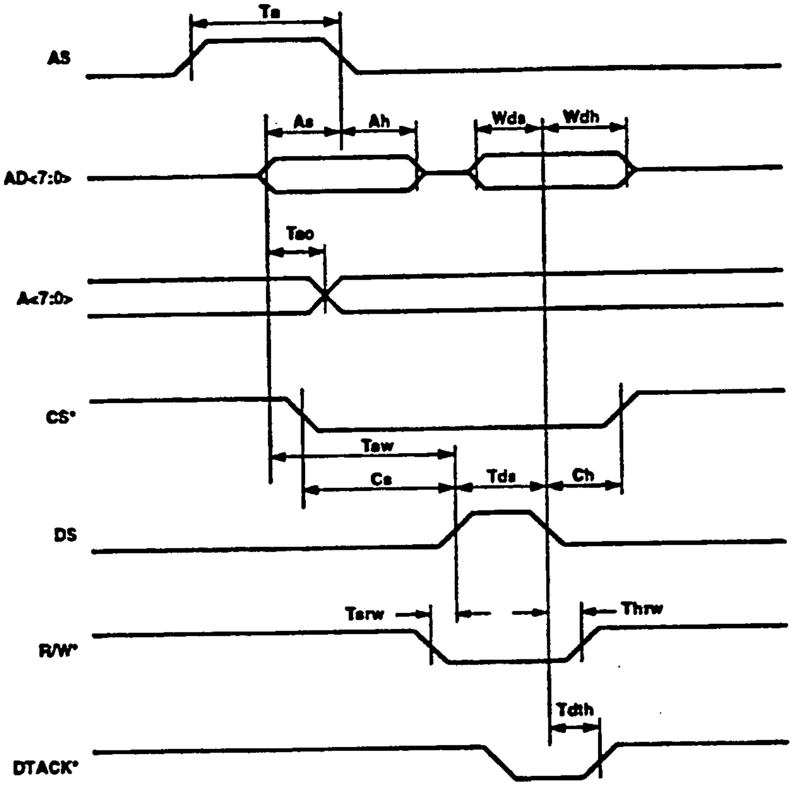
Note: If the DSH/L* input is '0' then the polarity of DS and AS is inverted.

9.3.2.6 Register Read Operation in Motorola Nonmultiplexed Mode



Note: If the DSH/L* input is '0' then the polarity of DS and AS is inverted.

9.3.2.7 Register Write Operation in Motorola Multiplexed Mode



Note: If the DSH/L* input is '0' then the polarity of DS and AS is inverted.

CIRRUS LOGIC

CL-SH4400

Sampled Amplitude Digital R/W Channel

The CL-SH4400 Sampled-Amplitude Digital Channel provides the digital core of a highly flexible, high-performance sampled-amplitude read/write channel which provides data rates up to 64 Mb/s. A companion analog part, the VM6400, implements the frequency synthesizer, variable gain amplifier, timing VFO, tunable analog filter, analog-to-digital converter, write precompensation, and servo demodulation functions. The CL-SH4400/VM6400 chip set supports hard-sectored magnetic disk drive applications employing embedded servo techniques and zone bit recording. The functional partition of the CL-SH4400/VM6400 chip set is shown in Figure 1.

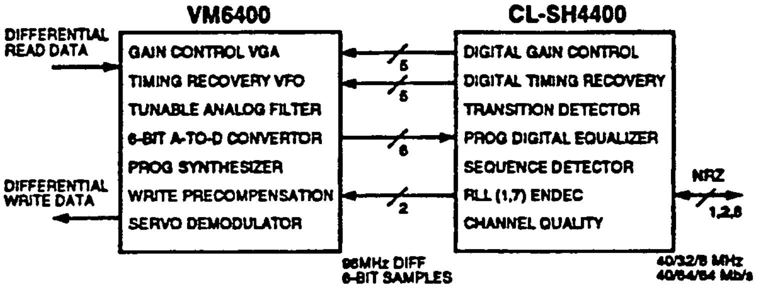


Figure 1 CL-SH4400/VM6400Functional Partition

The CL-SH4400 provides digital gain control, timing recovery, equalization, sequence detection, RLL (1,7) encoding and decoding, error-tolerant synchronization, and channel quality measurement. The high degree of programmability of each of these functions allows the CL-SH4400 to support highly adaptive channels tailored for each drive, head, and zone.

The format of a typical sector containing one embedded servo area is shown in Figure 2. During a read(write) operation, the Disk Controller reads the offset of the servo area from the ID. It deasserts Read(Write) Gate after the specified number of bytes have been processed and reasserts it after a programmed number of bytes equal to the length of the servo area. The VM6400 and external circuitry read and decode the servo bursts and servo data. During a read operation, the CL-SH4400 freezes its gain control and timing recovery states while Read Gate is deasserted.

PREAMBLE	SYNC	ID	CRC	SPLICE	PREAMBLE	SYNC	DATA	SERVO	SPLICE	PREAMBLE	SYNC	DATA	SPLICE

Figure 2 Typical Sector Format

CL-SH4400

Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

ADVANTAGES

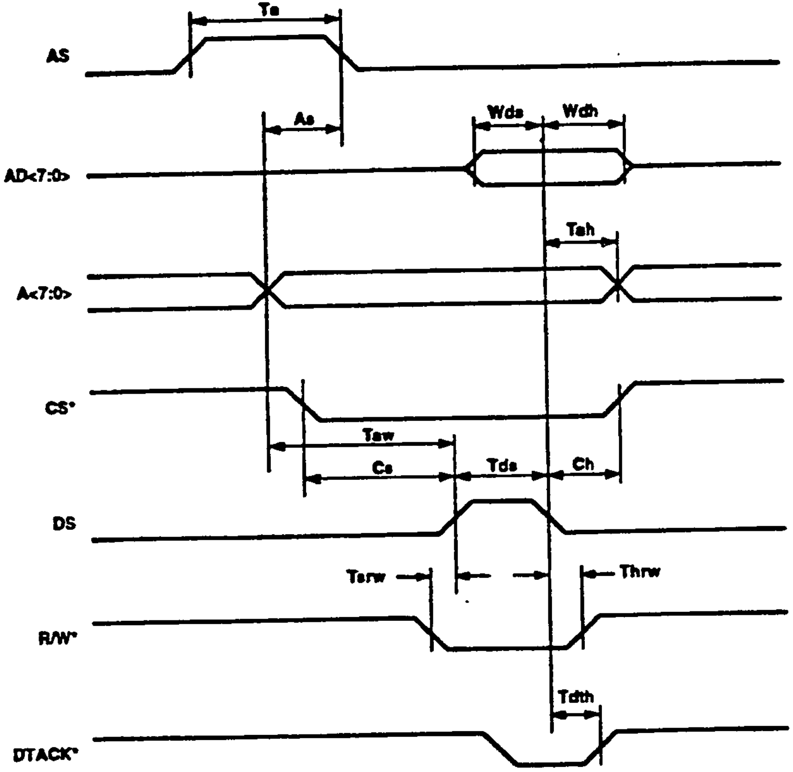
Benefits

- Flexible architecture implements broad range of partial response polynomials.
- Provides significant implementation and performance benefits over conventional peak detection.
- RLL (1,7) code reduces nonlinear effects associated with closely-spaced transitions.
- Programmable equalization, sampling, and detection alternatives provide optimum match to channel characteristics and permit implementation of adaptive channels.
- SofTarget™ sequence detection accommodates a lower signal-to-noise ratio than hard thresholding of individual samples and allows significant intersymbol interference without negative consequences.
- Detector supports a broad class of partial response channels including PR4 (1,7), EPR4 (1,7) and EEPR4 (1,7).
- Error tolerance features support the higher soft error rates and higher defect densities encountered at higher recording densities.
- One/Two/Eight-bit NRZ Data interface connects to a broad range of Disk Controllers.
- Intelligent power management minimizes operational and idle power consumption.

Key Features

- One-bit, two-bit, or eight-bit NRZ Data interface
- Rate 2/3 RLL (1,7) recording code
- Digital Gain Control
- Digital Timing Recovery
- Digital Equalizer
- SofTarget™ Sequence Detector
- Error-tolerant synchronization
- Programmable gain, timing, equalization, sampling, and detection alternatives
- Channel quality circuit for filter/equalizer/detector calibration and error rate estimation
- Erasure pointer generation
- Support for monolithic, composite, thin-film, MIG and magneto-resistive head technologies.
- Intelligent power management

9.3.2.8 Register Write Operation in Motorola Nonmultiplexed Mode



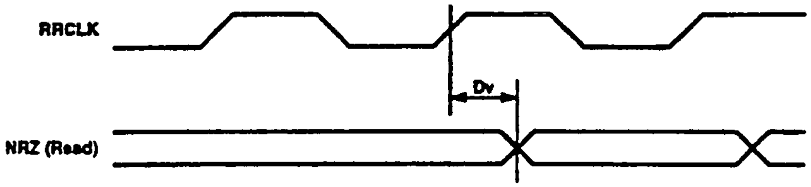
Note: If the DSH/L* input is '0' then the polarity of DS and AS is inverted.

9.3.3 Disk Read/Write Timing Parameters

One(Two) Bit NRZ Interface

Symbol	Parameter	Minimum	Maximum	Units
T	RRCLK period	25(31)		ns
T/2(L)	RRCLK low time at 0.8V	10(12)		ns
T/2(H)	RRCLK high time at 2.0V	10(12)		ns
Tr, Tf	RRCLK rise and fall time		3	ns
Ds	NRZ in valid to RRCLK ↑	5		ns
Dh	RRCLK ↑ to NRZ in invalid	5		ns
Dv	RRCLK ↑ to NRZ out valid	6	18	ns

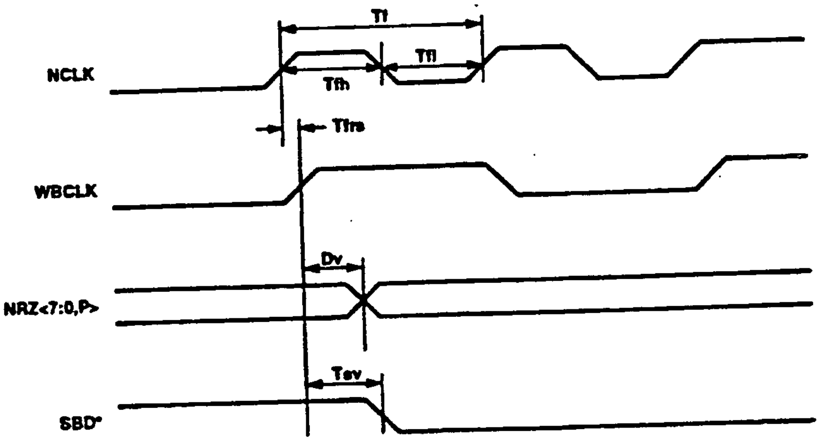
Note: ↑ indicates rising edge,↓ indicates falling edge.



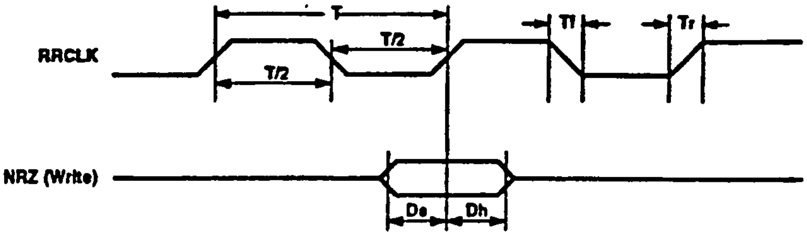
Eight Bit NRZ Interface

Symbol	Parameter	Minimum	Maximum	Units
Tw	WBCLK period	125		ns
Twl	WBCLK low time	15		ns
Twh	WBCLK high time	15		ns
Ds	NRZ in setup to WBCLK ↑	20		ns
Dh	WBCLK ↑ to NRZ in hold	20		ns
Pv	WBCLK ↑ to WPERR valid	20		ns
Ph	WBCLK ↓ to WPERR hold	20		ns
Tf	NCLK period	62		ns
Tfl	NCLK low time	20		ns
Tfh	NCLK high time	20		ns
Tfrs	NCLK to RBCLK skew	-5	5	ns
Tr	RBCLK period	125		ns
Dv	RBCLK ↑ to NRZ valid	5	20	ns
Tsv	RBCLK ↑ to SBD* valid	5	20	ns

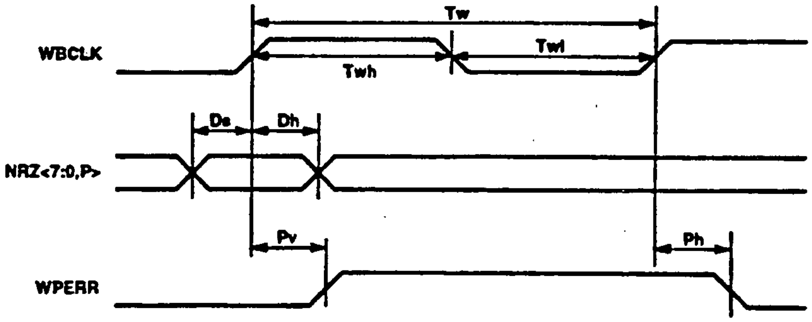
Note: ↑ indicates rising edge,↓ indicates falling edge.



9.3.3.1 Disk Write Timing (One/Two Bit NRZ Interface)



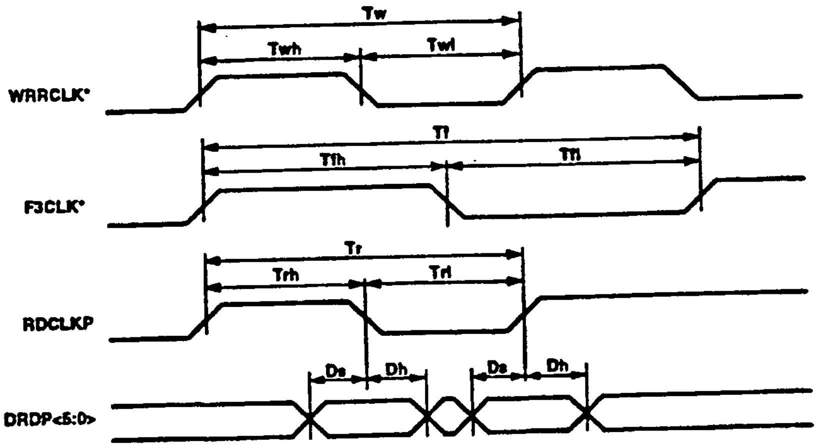
9.3.3.2 Disk Write Timing (Eight Bit NRZ Interface)



9.3.4 VM6400 Interface Timing Parameters

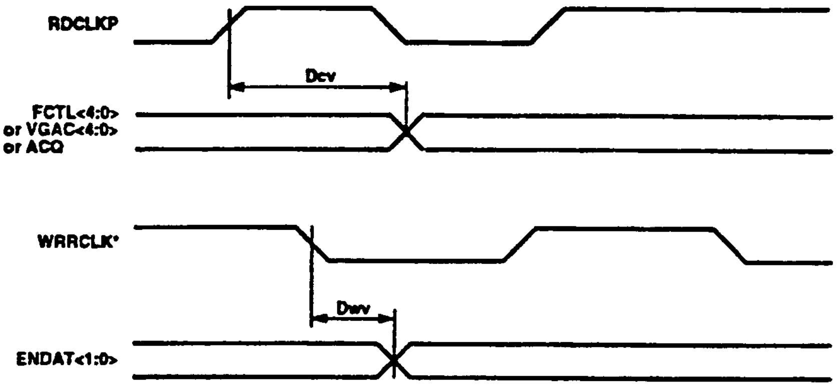
Input

Symbol	Parameter	Minimum	Maximum	Units
Tw	WRRCLK* period	20		ns
Twl	WRRCLK * low time	8		ns
Twh	WRRCLK* high time	8		ns
Tf	F3CLK* period	31		ns
Tfl	F3CLK* low time	12		ns
Tfh	F3CLK* high time	12		ns
Tr	RDCLKP/N period	20		ns
Trl	RDCLKP/N low time	9		ns
Trh	RDCLKP/N high time	9		ns
Ds	DRDP/N setup time	3		ns
Dh	DRDP/N hold time	5		ns



Output

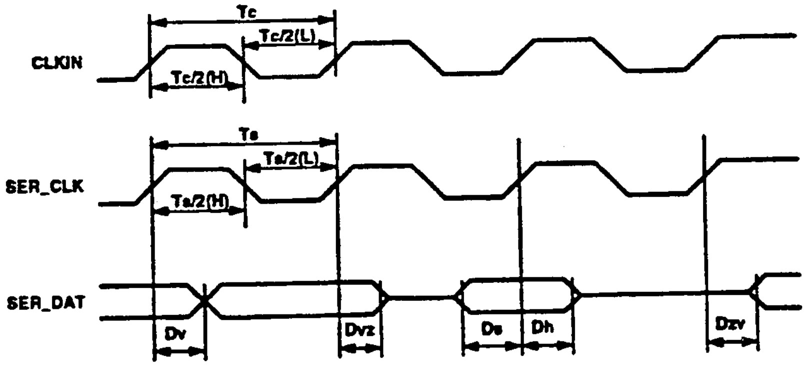
Symbol	Parameter	Minimum	Maximum	Units
Dcv	RDCLKP ↑ to FCTL/VGAC/ACQ valid	3	15	ns
Dwv	WRRCLK* ↓ to ENDAT valid	3	15	ns



9.3.5 Serial Control Interface Timing Parameters

Symbol	Parameter	Minimum	Maximum	Units
Tc	CLKIN period	25		ns
Tc/2(L)	CLKIN low time	8		ns
Tc/2(H)	CLKIN high time	8		ns
Ts	SER_CLK period	40		ns
Ts/2(L)	SER_CLK low time	15		ns
Ts/2(H)	SER_CLK high time	15		ns
Dv	SER_CLK ↑ to SER_DAT valid		15	ns
Dvz	SER_CLK ↑ to SER_DAT undriven		20	ns
Dzv	SER_CLK ↑ to SER_DAT driven		20	ns
Ds	SER_DAT ↑ to SER_CLK setup time	10		ns
Dh	SER_DAT ↑ to SER_CLK hold time	10		ns

9.3.5.1 Serial Control Timing



6.6 0AH Data Randomizer Seed LSB

Bits 7-0 **Data Randomizer Seed LSB:** These are the eight least-significant bits of the Data Randomizer seed.

6.7 0BH Data Randomizer Seed MSB

Bits 7-0 **Data Randomizer Seed MSB:** These are the eight most-significant bits of the Data Randomizer seed.

6.8 10H Acquisition Length

Bit 7 **Pulse Shape Compensation Disable:** This bit controls whether the Gain Control and Timing Recovery circuits use Detector Levels *a*, *b*, and *c* to compensate for intersymbol interference. This bit does not affect the use of Detector Levels *a*, *b*, and *c* by the SofTarget™ Sequence Detector.
0: Pulse Shape Compensation is enabled.
1: Pulse Shape Compensation is disabled.

Bit 6 **Header Precursor ISI Compensation Disable:** This bit controls whether the Spectrum Smoothing Filter compensates for precursor intersymbol interference during headers. Setting this bit has the effect of forcing SD1 to zero when Sequencer Output is asserted; SC1 and SC2 are ignored and the pipeline delay of the Spectrum Smoothing Filter is minimized. This may be useful for decreasing the delay between DRD input and NRZ output for headers, at the cost of a small increase in error rate.
0: Header Precursor ISI Compensation is enabled
1: Header Precursor ISI Compensation is disabled.

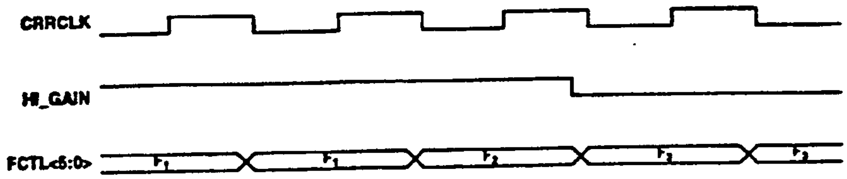
Bits 5-0 **Acquisition Length:** This value controls the duration, in increments of four channel-bit periods, of acquisition for Gain Control and Timing Recovery. The minimum value which should be written here is 12, which corresponds to 48 channel-bit periods. The maximum value which should be written here should correspond to at least 12 channel-bit periods less than the total length of the preamble excluding the length of the Synchronization Mark. The total length of the preamble is determined by the Disk Controller during write.

6.9 11H Acquisition Gain Set Point

Bits 7-6 **Transition Threshold Bits 1-0:** These bits specify the two least-significant bits of the four-bit Transition Threshold (V) used to qualify transitions during tracking.

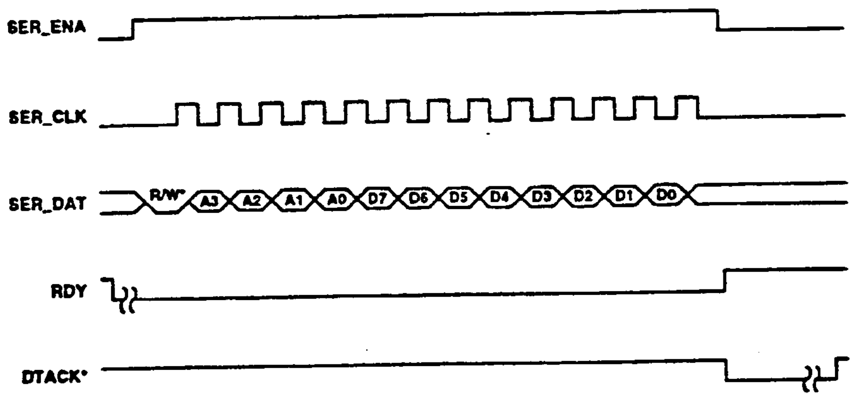
Bits 5-0 **Acquisition Gain Set Point** These bits specify the value (g_a) used in computing gain errors during acquisition. Range is 0 to +63/16.

9.3.6.3 Timing Recovery Operation

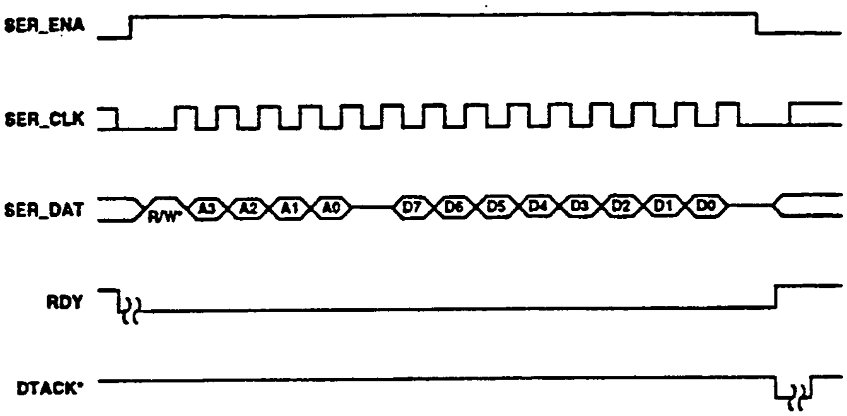


Note: F_1 = Direct path + Acquisition Timing Recovery Loop Accumulator
F_2 = Acquisition Timing Recovery Loop Accumulator
F_3 = Direct path + Tracking Timing Recovery Loop Accumulator

9.3.6.4 Serial Control Write Register Operation



9.3.6.5 Serial Control Read Register Operation



10. GLOSSARY

Aliasing - The phenomenon whereby energy at frequencies above one-half the sampling rate appears as energy at frequencies below one-half the sampling rate in a sampled signal.

Antialiasing Filter - A filter which attenuates signal energy above a frequency of one-half the sampling rate.

Autocorrelation Function - The integral $\phi(\tau)$ of the product of a function $x(t)$ and $x(t+\tau)$; e.g $\phi(\tau) = E\{x(t)\cdot x(t+\tau)\}$. For a binary sequence, this can be treated as the number of bits matching between two copies of a sequence as a function of the offset between the sequences.

Calibration - The process whereby the analog and digital parameters of the channel and the detector are matched for best performance.

Channel Constraint - A limitation placed on the sequence of channel bits. For example, an RLL (d,k) channel constraint enforces a minimum (d) and maximum (k) number of consecutive '0's between '1's.

Channel Quality - A measure of how well matched are a channel and detector.

Channel Code - A set of rules goveming the encoding of data bits to channel bits. Channel codes typically enforce a minimum run-length constraint (to limit non-linear effects) and/or a maximum run-length constraint (to aid in timing recovery).

Code Rate - The ratio of the number of bits in the data-bit stream to the number of bits in the channel-bit stream. For an RLL (1,7) code, the code rate is 2/3.

Coding Gain - The product of the code rate and the minimum distance of the encoded sequences of a code.

Convolution - The convolution y(k) of sequences u(k) and h(k) is given by

$$y(k) = \Sigma\, u(m)\ \ h(k - m)$$

where the summation is over all m. When u(k) is the input to a filter whose impulse response is h(k), the filter output is the convolution y(k).

Correlation Function - The expected value of the product of two random variables, which may be scalars or vectors.

Dicode - Another name for Partial Response Class I (PR1).

Distance - A measure of the "difference" between two vectors.

Duobinary - The partial response channel corresponding to the polynomial (1 + D).

Error Propagation - The process wherein a channel decoder emits more than one erroneous data bit due to a single channel bit in error or more generally a burst of erroneous data bits which is longer (after accounting for code rate) than the causative channel-bit error burst.

Error Tolerant Synchronization Mark - A synchronization mark which can be detected in the presence of error with an acceptable false detection probability.

Euclidean Distance - The square root of the sum of the squares of the differences in each dimension between two n-dimensional vectors.

Excess Bandwidth - Signal energy passed by a filter at frequencies above one-half the sampling rate.

Extended Partial Response Class IV (EPR4) - The partial response channel corresponding to the polynomial $(1 - D)(1 + D)^2$.

Extended Extended Partial Response Class IV (EEPR4) - The partial response channel corresponding to the polynomial $(1 - D)(1 + D)^3$.

Eye Diagram - A graphical representation of the effects of intersymbol interference. It illustrates the typical or maximum excursion of a signal from its ideal sample levels at its ideal sampling instants. It is important to note that Viterbi Sequence Detection is not dependent on the eye being open in the same way that sample-by-sample detection is.

Least-Mean-Square Algorithm - A procedure used in adapting the coefficients of a digital filter in which the objective is to minimize the mean-square error of the filter output.

Matched Filter - A filter whose impulse response is the time reversal of the channel to which it is connected.

Maximum A Posteriori Probability Sequence Detection (MAP) - A decoding method in which the channel bit sequence emitted from the detector is that which, given the channel sample sequence, is the most probable written sequence. MAP maximizes the probability density function of the detector output given the detector input.

Maximum Likelihood Sequence Detection (MLSD) - A decoding method in which the channel bit sequence emitted from the detector is that with the highest probability of having produced the channel sample sequence input to the detector. MLSD maximizes the probability density function of the detector input given any possible channel input. MLSD is equivalent to MAP if the probabilities of all possible written sequences are equal or if they are assumed to be equal when no information is available concerning the probabilities of written sequences.

Mean-Square Error - A measure of error in which the difference between each sample and its ideal sample level is squared and summed. The sum divided by the number of samples is the mean-square error of the sequence.

Metric - A function of two vectors satisfying certain properties that allow it to be interpreted as a measure of distance.

Minimum Distance - The smallest distance between any two vectors in a set of vectors.

Modified Duobinary - Another name for Partial Response Class IV (PR4).

Noise Types:

Coherent Noise - Noise which is correlated with the signal; e.g. transition noise.

Colored Noise - Noise which is correlated between one sample instant and another.

Gaussian Noise - Noise whose amplitude has a normal probability distribution.

Misequalization Noise - Amplitude errors which remain at the output of an equalizer due to its imperfect pulse-shaping. This is something of a misnomer, as this "noise" source is predictable for a given channel transition response, equalizer, and transition sequence.

Offtrack Noise - When tracking is not perfect and the head deviates from the center of a track, the amplitude of the signal from the target track is reduced and the head picks up signal energy due to transitions on the adjacent track.

Out-of-band Noise - Noise with energy at frequencies above one-half the sampling rate.

Overwrite Noise - In thin film media the magnetic fields of previously written transitions affect the location and shape of transitions written over them.

Preamp Noise - Electronic noise arising in the first stage of the head preamplifier. Due to the low signal amplitude present at the input to the preamp and the correspondingly high gain required, it is possible for preamp noise to be the dominant noise source in a read channel.

Quantization Noise - Error introduced by a finite representation of a real number through A/D conversion or internal rounding.

Synchronous Noise - An interfering source with a frequency equal to the sample rate or a multiple thereof. The energy of such a source is aliased to DC and can be canceled by adjusting the VGA DC Offset of the VM6400.

Timing Error - Incorrect timing of the aid sampling process due to VFO jitter and other noise passed through the timing control loop.

Transition Noise - Coherent noise caused by "zig-zag" transitions occurring on thin-film media.

White Noise - Noise which is not correlated between one sample instant and another.

Nonlinear Transition Shift - Movement in the effective location of a transition toward a very near preceding transition. This is caused by demagnetization during write.

Norm - A particular method of measuring the lengths of vectors; related to a metric.

NRZ - Non-Return to Zero refers to a bit sequence to which no run-length constraint applies.

NRZI - Modified Non-Return to Zero refers to a bit sequence to which no run-length constraint applies but in which each '1' or '0' represent the presence or absence, respectively, of a transition. NRZI sequences can be produced from NRZ sequences via a precoding process equivalent to dividing by (1 + D).

Partial Response Channel - A channel in which controlled intersymbol interference is allowed. In such a channel, symbols (e.g. transitions in a magnetic recording channel) are so closely spaced that the channel begins to respond to the next symbol when only part of its response to the current symbol has occurred.

Partial Response Class I (PR1) - The partial response channel corresponding to the polynomial (1 - D).

Partial Response Class IV (PR4) - The partial response channel corresponding to the polynomial

(1 - D)(1 + D).

Path Metric - A cumulative measure of error maintained for a path through a decoder's state machine based on a particular norm.

Preamble - A special repetitive pattern which is used to allow gain to be adjusted to an appropriate level and frequency and phase lock to be acquired. The CL-SH4400 can use a preamble pattern of "1010..." or "100100..." or "100010000...".

Precoding - The process of encoding a sequence to account for some known property of a channel e.g. the differentiating nature of a magnetic recording channel.

Raised Cosine - A form of equalization which is a more-realizable modification of the "brick wall" filter which would produce a sinc pulse. It is characterized by a transfer function H(f) defined as:

$$\begin{aligned} H(f) &= 1 \text{ for } 0 < f < (1 - \alpha)f_S/2 \\ &= (1/2)[1 + \cos(\pi/[\alpha f_S][f - (1 - \alpha)f_S/2])] \text{ for } (1 - \alpha)f_S/2 \le f \le (1 + \alpha)f_S/2 \\ &= 0 \text{ for } f > (1 + \alpha)f_S/2 \end{aligned}$$

where f_S is the sample rate and α is a parameter in the range zero to one.

Run-Length Limited Code - A code in which the minimum and maximum numbers of consecutive '0's and/ or '1's is constrained by a systematic rule. The notation RLL (d,k) means that no less than d and no more than k consecutive '0's are allowed in the encoded bit stream, where '0' and '1' usually represent the absence or presence, respectively, of a (magnetic) transition.

Run-Length Violation - An case in which a defect or noise event has caused the occurrence of a sequence of '0's and '1's which violates the run-length constraints of the code.

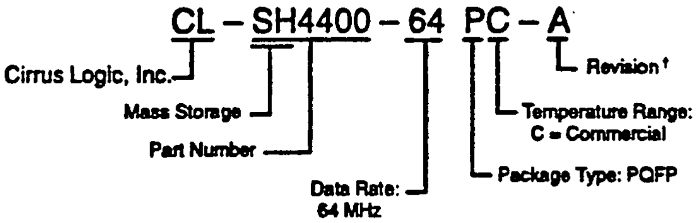
Split Field - A recording scheme in which the data/redundancy area of a sector is divided into two or more areas, typically by one or more areas used for tracking control in an embedded-servo disk drive.

State Machine - A circuit in which the present state of a channel or system is represented by a number of bits whose values are controlled by the present values of the bits and the values of external signal levels or the occurrence of external signal transitions.

Synchronization Mark - A pattern, preferably unique and different from any valid encoded channel-bit sequence, used to signify the start of a header or data area. A good synchronization mark has low cross-correlation with the preamble and small autocorrelation function values at non-zero offsets, implying a low false detection probability.

Viterbi Sequence Detector - A sequence detector which is a maximum-likelihood sequence detector under the conditions of matched-filtered input and additive white Gaussian noise.

11. ORDERING INFORMATION



[†] Contact Cirrus Logic, Inc. for up-to-date information on revisions.

CL-SH3300
Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

SH3300 SAMPLED AMPLITUDE DIGITAL READ CHANNEL

Product Description

The SH3300 sampled amplitude synchronous channel is a mixed-signal CMOS integrated circuit that implements a highly flexible synchronous channel.

Partial response schemes such as PR4 and EPR4 may be implemented, and the SofTarget ™ Sequence detector provides a user specific partial response scheme that allows the detection strategy to be customized for a specific pulse shape.

Additionally, the degree of flexibility in the SH3300 architecture allows for more precise matching of HDA and channel electronics.

The SH3300 offers a wide array of power down modes, allowing the disk drive designer to achieve extremely low average power consumption.

The SofTarget ™ technology, together with an on-chip filter specifically designed to minimize pole-tip effects or secondary gap effects make the SH3300 compatible with monolithic, composite, thin film, MIG and magneto-resisitive heads.

FEATURES

The SH3300 provides a highly flexible, sampled-amplitude read/write channel at channel rates up to 60 MHz .

Complete Read Channel Detector

Rate 2/3 RLL (1,7) Recording Code employing SofTarget ™ Detection Technology

Completely implements synchronous channel with only 4 off-chip non-precision passives

VGA with digital gain control

7th order Equi-ripple phase filter with variable cut-off and variable boost using two symmetrical real axis zeros

Programmable sequence detection

Digital timing recovery and offset control

Error Tolerance

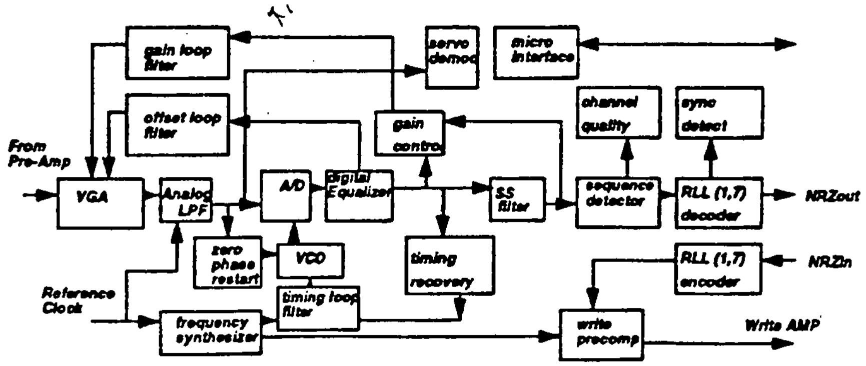
Error-tolerant synchronization

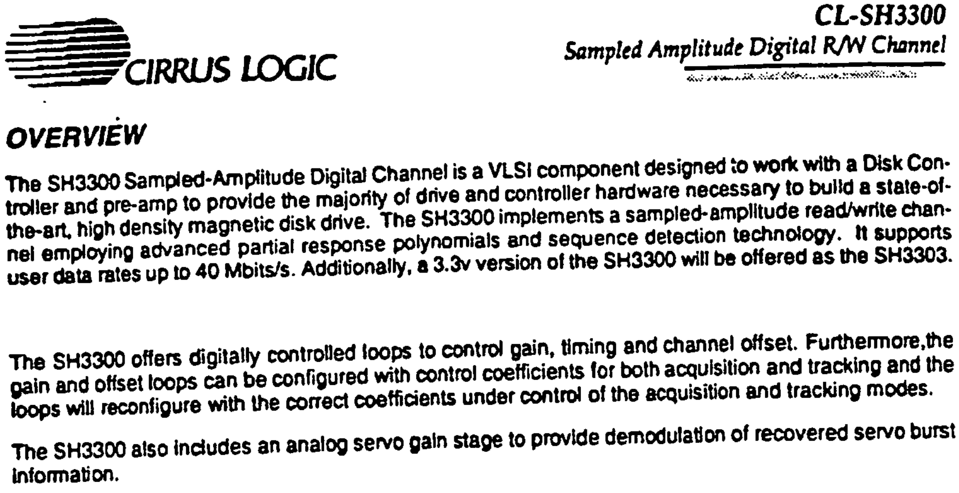
Channel quality circuitry for error rate testing and channel calibration

Technology

0.8 micron double metal CMOS / 64 pin VQFP package

Functional Block Diagram





CIRRUS LOGIC

CL-SH3300

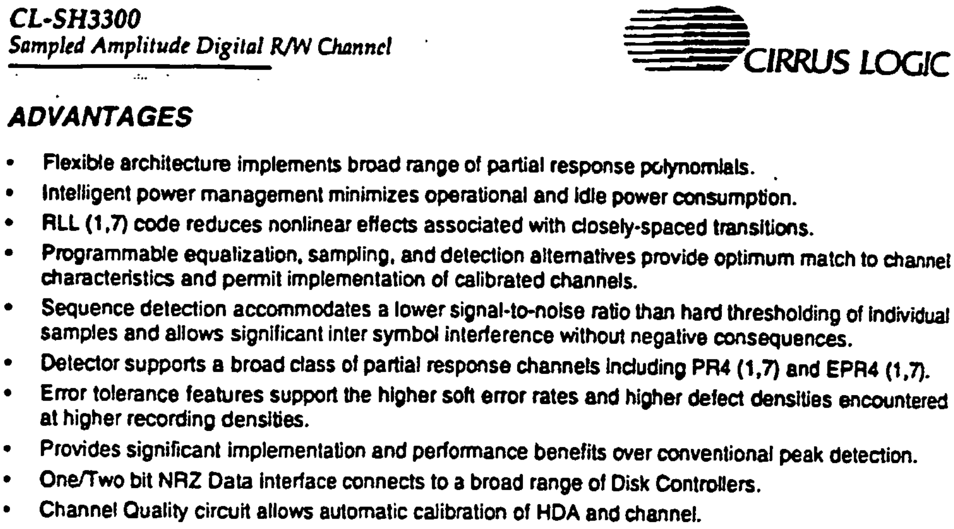
Sampled Amplitude Digital R/W Channel

OVERVIEW

The SH3300 Sampled-Amplitude Digital Channel is a VLSI component designed to work with a Disk Controller and pre-amp to provide the majority of drive and controller hardware necessary to build a state-of-the-art, high density magnetic disk drive. The SH3300 implements a sampled-amplitude read/write channel employing advanced partial response polynomials and sequence detection technology. It supports user data rates up to 40 Mbits/s. Additionally, a 3.3v version of the SH3300 will be offered as the SH3303.

The SH3300 offers digitally controlled loops to control gain, timing and channel offset. Furthermore,the gain and offset loops can be configured with control coefficients for both acquisition and tracking and the loops will reconfigure with the correct coefficients under control of the acquisition and tracking modes.

The SH3300 also includes an analog servo gain stage to provide demodulation of recovered servo burst information.



CL-SH3300
Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

ADVANTAGES

- Flexible architecture implements broad range of partial response polynomials.
- Intelligent power management minimizes operational and idle power consumption.
- RLL (1,7) code reduces nonlinear effects associated with closely-spaced transitions.
- Programmable equalization, sampling, and detection alternatives provide optimum match to channel characteristics and permit implementation of calibrated channels.
- Sequence detection accommodates a lower signal-to-noise ratio than hard thresholding of individual samples and allows significant inter symbol interference without negative consequences.
- Detector supports a broad class of partial response channels including PR4 (1,7) and EPR4 (1,7).
- Error tolerance features support the higher soft error rates and higher defect densities encountered at higher recording densities.
- Provides significant implementation and performance benefits over conventional peak detection.
- One/Two bit NRZ Data interface connects to a broad range of Disk Controllers.
- Channel Quality circuit allows automatic calibration of HDA and channel.

CIRRUS LOGIC

CL-SH3300
Sampled Amplitude Digital R/W Channel

TABLE OF CONTENTS

CL-SH3300
Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

CIRRUS LOGIC

CL-SH3300
Sampled Amplitude Digital R/W Channel

1. INTRODUCTION

1.1 Overview

The SH3300 Sampled-Amplitude Digital Channel provides a highly flexible, highly integrated sampled amplitude read/write channel which supports NRZ data rates up to 40 MBits/sec. The SH3300, implements the frequency synthesizer, variable gain amplifier, tunable analog filter, analog-to-digital converter, write precompensation, and servo demodulation functions in the analog domain, and provides digital gain and offset control, timing recovery, equalization, sequence detection, RLL (1,7) encoding and decoding, error-tolerant synchronization, and channel quality measurement in the digital domain.The SH3300 is compatible with hard-sectored magnetic disk drive applications employing embedded servo techniques.

The high degree of programmability of each of these functions allows the SH3300 to support highly adaptive channels tailored for each drive, head, and zone.

The format of a typical sector containing one embedded servo area is shown in Figure 1.1-1. During a read(write) operation, the Disk Controller reads the offset of the servo area from the ID. It deasserts Read(Write) Gate after the specified number of bytes have been processed and reasserts it after a programmed number of bytes equal to the length of the servo area. The SH3300 , together with an external servo decoder reads and decode the servo bursts and servo data.

PREAMBLE	SYNC	ID	CRC	SPLICE	PREAMBLE	SYNC	DATA	SERVO	SPLICE	PREAMBLE	SYNC	DATA	SPLICE

Figure 1.1-1. Typical Sector Format

The format of a typical six-byte ID is shown in Figure 1.1-2. The Flag/MS_Split byte contains bits controlling split-field operations, end-of-track, and defective sector, and the most-significant bits of the split field count. The LS_Split byte contains the least-significant bits of the split field count. The split field count specifies the number of bytes between the start of data and the start of the embedded servo area.

CYLINDER HIGH	CYLINDER LOW	HEAD	SECTOR	FLAG/MS_SPLIT	LS_SPLIT

Figure 1.1-2. Typical ID Format

1.2 Key Features

- One-bit or two-bit NRZ Data interface
- Rate 2/3 RLL (1,7) recording code
- Digital Gain Control
- Digital Timing Recovery
- Digital Offset Control
- Digital Equalizer
- Sequence Detector
- Digitally controlled VGA with separate offset cancellation control
- 7th order Equi-ripple filter with programmable cut-off and boost

CL-SH3300

Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

- 60Msamples/s 6-bit flash A to D converter
- Write precompensation support for 101 pattem
- Standard servo peak demodulator supports up to 4 servo bursts
- Error-tolerant synchronization
- Programmable gain, timing, offset, equalization, sampling, and detection altematives
- Channel quality circuit for channel calibration and error rate estimation
- Erasure pointer generation
- Support for monolithic, composite, thin-film, MIG and magneto-resistive head technologies
- Intelligent power management

2. PIN INFORMATION

The following conventions are used in the pin assignment tables. Asterisk (*) denotes a negative true signal. (I) indicates an input pin. (O) indicates an output pin. (OD) indicates an open-drain output pin. (I/O) indicates a bidirectional pin. All unused inputs must be tied to the inactive state, VDD or GND.

2.1 Microcontroller Interface Pins

SYMBOL	PIN	TYPE	DESCRIPTION
RST*		I	RESET: Assertion of this signal stops all operations within the SH3300 and de-asserts all output signals. All input/output signals are set to the high-impedance state.
CS*		I	CHIP SELECT: This signal must be asserted for all microcontroller accesses to the SH3300's registers.
I/MC*		I	INTEL/MOTOROLA: This signal selects the microcontroller interface to be used. When this signal is high, it selects the Intel bus control interface. When this signal is low, it selects the Motorola bus control interface. An internal pullup allows this signal to be legally "floated" to select the default Intel bus control interface.
WR*/R-W*		I	WRITE STROBE/READ_WRITE: When I/MC* is high, this signal is WR*; when CS* and WR* are asserted, the data on the AD bus will be written to the specified register. When I/MC* is low, this signal is R-W*, which determines the direction of data transfer when accessing the SH3300's registers. When CS* and DS are asserted and R-W* is high, a register read operation is in progress. When CS* and DS are asserted and R-W* is low, a register write operation is in progress.
ALE/AS		I	ADDRESS LATCH ENABLE/ADDRESS STROBE: On the trailing edge of this signal, the SH3300 latches the address present on the AD or A bus, as selected by M/NM*.
RD*/DS		I	READ STROBE/DATA STROBE: When I/MC* is high, this signal is RD*; when CS* and RD* are asserted, the data from the specified register will be driven onto the AD bus. When I/MC* is low, this signal is DS, which determines the data timing of a register access. When CS* is asserted and R-W* is high, the rising edge of DS indicates when the SH3300 may start driving data onto the AD bus. When CS* is asserted and R-W* is low, the trailing edge of DS indicates when the SH3300 may latch data from the AD bus.
M/NM*		I	MULTIPLEXED/NONMULTIPLEXED: When this signal is high, AD<7:0> is the multiplexed address/data bus and A<7:0> is the latched address output bus. When this signal

			is low, AD<7:0> is the data bus and A<7:0> is the address input bus. An internal pullup allows this signal to be legally "floated" to select the default multiplexed bus mode of operation.
RDY/DTACK*		OD	READY/DATA ACKNOWLEDGE: When I/MC* is high, this signal is the microcontroller Ready line; when this signal is deasserted, the local microcontroller shall insert wait states to allow time for the SH3300 to complete the requested access. When I/MC* is low, this signal is the Data Acknowledge signal; this signal is asserted when the SH3300 has completed the requested access.
AD<7:0>		I/O	MICROCONTROLLER ADDRESS/DATA BUS: When M/NM* is high, AD<7:0> is the multiplexed address/data bus. When M/NM* is low, AD<7:0> is the data bus.
A<7:0>		I/O	MICROCONTROLLER ADDRESS: A<7:0> iis the address input bus used to determine the register location to be accessed. In demultiplexed mode register location should be input on A<0:7>. In multiplexed mode A<7:0> are not used and should be tied to ground. Note that in multiplexed mode the SH3300 does not provide a a latched address output.
MCLK		I	MASTER REFERENCE CLOCK: This input is used to derive the synthesized clock output and WCLK during a write

2.2 Preamp Interface Pins/Diagnostics

SYMBOL	PIN	TYPE	DESCRIPTION
IN+		I	Positive input from pre-amp
IN-		I	Negative input from pre-amp
WDO		O	Positive output to write amp
P CS		O	pre-amp chip select
P R/W		O	pre-amp read/write
DOUT +		O	Under register control this pin, together with DOUT- will provide VGA outputs or filter outputs for development purposes. A register bit sets the source for this output.
DOUT-		O	See DOUT+

2.3 Servo Interface Pins

SYMBOL	PIN	TYPE	DESCRIPTION
SVENC<3:0>		I	SERVO ENCODE: Encoded servo command field. This field contains encoded servo commands for gating the burst sample and holds, indicating the presence of the servo fields, causing the servo sample and holds to reset etc. See the section on servo operation for a list of encodes..
SVCLK		I	SERVO CLOCK: Used to clock the SVENC bits into Lexington.
SBA/SAMB		O	SERVO BURST A OUT: Peak detected A burst or, under register control, will output A-B
SBB		O	SERVO BURST B OUT: Peak detected B burst
SBC/SCMD		O	SERVO BURST C OUT: Peak detected C burst, or under register control will output C-D
SBD		O	SERVO BURST D OUT: Peak detected D burst
SBIT		O	SERVO BIT : will output peak detected servo information out of SH3300. Typically, this information would include servo preamble, servo address mark, servo sync mark, index and track number
SBITP		I	SERVO BIT POLARITY: determines if the transition that caused the SBIT output is positive or negative.
SVREF		I/O	SERVO REFERENCE: This pin serves either as an input for an external reference or it will provide a reference that can be used externally. The ABCD values will be referenced to this voltage and will always be above this voltage. If the burst output pins are used in the A-B and C-D modes, then the levels on these pins will be above or below the reference.

2.4 Disk Controller Interface Pins

SYMBOL	PIN	TYPE	DESCRIPTION
RG/RG*		I	READ GATE: When this signal is asserted, the read path circuitry is enabled.
WG/WG*		I	WRITE GATE: When this signal is asserted, the write path circuitry is enabled.
NRZ<0>		I/O	NON-RETURN TO ZERO BIT 0: In One-Bit NRZ Mode, this pin serially transmits(receives) the least-significant

		unencoded read(write) data bit to(from) the Disk Controller. In Two-Bit NRZ Mode, this signal transmits(receives) the least-significant unencoded read (write) data bit to(from) the Disk Controller. This pin is in the high impedance mode if RG/RG* is inactive.
NRZ<1>	I/O	**NON-RETURN TO ZERO BIT1:** In One-Bit NRZ Mode, this pin is driven into the high-impedance mode. In Two-Bit NRZ Mode, this signal transmits(receives) the most-significant unencoded read (write) data bit to(from) the Disk Controller. This pin is in the high impedance mode if RG/RG* is inactive.
RRCLK	O	READ REFERENCE CLOCK: This signal is used to clock NRZ to and from the Disk Controller. During a disk read operation, the RRCLK signal is used by the disk controller to strobe read data from the SH3300 into the controller. During a write operation, the RRCLK signal is used to clock data out of the disk controller.
WCLK	I	WRITE CLOCK: During a disk write operation this signal is used to clock data into the SH3300. WCLK can either be generated directly from the RRCLK output tied back into the WCLK input, or from a WCLK if one is supplied by the disk controller.
SEQ_OUT	I	**SEQUENCER OUTPUT:** This signal is connected to the Sequencer Output of the Disk Controller. It is used to control the Synchronization Mark Recovery Mode and the Path Memory Length.

2.5 Power, Ground, and Miscellaneous Pins

SYMBOL	PIN	TYPE	DESCRIPTION
VDD		N/A	DIGITAL POWER SUPPLY
DGND		N/A	DIGITAL GROUND
VDDA		N/A	ANALOG POWER SUPPLY
AGND		N/A	ANALOG GROUND

2.6 Pin Diagram

64 Pin VQFP pinout is <TBD>.

3. REGISTER TABLES

<TBD>

4. FUNCTIONAL DESCRIPTION

The major functional blocks of the SH3300 include:

- NRZ Data Interface
- Transition Detector
- Gain Control and Variable Gain amp
- Timing Recovery and VCO
- Flash A/D converter
- Digital Equalizer/Spectral Smoothing Filter
- Analog equalizer
- Sequence Detector
- Synchronization Mark Detector
- RLL Encoder/Decoder
- Serial Control Interface
- Channel Quality

4.1 Write Operation

When Write Gate is asserted, the Disk Controller must first emit a number of bytes on the NRZ Data interface equal to the desired length of the preamble. When fed with repeating hex '88', 'FF', or 'D3' NRZ bytes, the RLL Encoder in the SH3300 encodes them to create a '1010...', '100100...', or '10001000...' preamble pattern, respectively, which it emits on the Encoded Data interface. After it has transferred the desired number of preamble bytes, the Disk Controller must emit a number of hex '00' bytes on the NRZ Data interface equal to the Synchronization Mark Length. After it has transferred the proper number of synchronization bytes, the Disk controller must emit data bytes on the NRZ Data interface. If the Data Randomizer is enabled, the data bytes are randomized to equalize the probability of occurrence of worst-case patterns. The RLL Encoder in the SH3300 encodes the data bits and the encoded data bits are emitted on the Encoded Data interface. The preamble bits, Synchronization Mark bits, and encoded data bits are all processed through the SH3300 write precompensation circuitry and transmitted to the write head.

4.1.1 Write Precompensation

Due to the nature of the Sequence Detector, which takes linear intersymbol interference into account when calculating path metrics, write precompensation for linear intersymbol interference is not required. However, demagnetizing effects during write can cause a shift in the location of a transition which is very close to the preceding transition. The SH3300 provides write precompensation to counter this effect. It examines the sequence of RLL-encoded bits for '101' and retards the second transition in all such sequences by a programmable amount.

4.2 Read Operation

During a read operation, the SH3300 passes the signal from the read/write preamplifier through a variable-gain amplifier and a tunable active filter, then generates digital samples from a six-bit flash analog-to-digital converter. The samples are fed to the programmable digital equalizer which further conditions the signal. The sample points of the analog to digital converter are determined by an oscillator intemal to the SH3300 whose frequency/phase is controlled to maintain proper sampling. The digitally-equalized samples are fed to the Sequence Detector in the SH3300. The output of the Sequence Detector is used to detect the Synchronization Mark and to feed the RLL Decoder, whose output is the NRZ data which is transmitted to the Disk Controller on the NRZ Data interface.

4.2.1 Number Representation and Arithmetic

The two's complement number system is used wherever negative and positive numbers must both be represented. The output signal samples from the analog-to-digital converter are six-bit twos complement numbers interpreted as having a binary point after the second most significant bit: SX.XXXX. Thus the range of representable numbers is from -2 to +31/16 in steps of 1/16. Note that the binary point is "virtual", having no existence in silicon.

Rounding and truncation are performed as considered appropriate in each Intemal operation. In some cases, intermediate results are carried with greater precision than the final result. Overflow is guarded against by saturating summers, by algorithmic bounds, or by probability of occurrence as appropriate in each intemal operation.

4.3 NRZ Data Interface

The SH3300 NRZ interface can be operated in one bit or two bit NRZ mode, via the NRZ<0> and NRZ,1> lines. In one bit mode the NRZ<0> line is used, and the NRZ<1> line is in a high impedance mode. . The One-bit NRZ Mode allows the SH3300 to connect to a disk controller which uses a one-pin serial NRZ data interface. When in the One-Bit NRZ read Mode, the SH3300 outputs a 2/3 channel rate clock (RRCLK) synchronous to the recovered read data. This RRCLK signal is typically connected to the RRCLK input of the disk controller. In two-bit NRZ read mode, the RRCLK is a 1/3 channel rate clock, and two bits of NRZ data (on NRZ<1> and NRZ<0> are clocked each cycle.

A WCLK signal input is provided to clock write data into the SH3300 during a write cycle. The WCLK signal be used in two ways.

First, it can be connected to the RRCLK developed by the SH3300. It is the resonsibility of the designer to ensure that the time taken for the disk controller to output write data (relative to the RRCLK input), does not violate the write data in to RRCLK setup and hold specifications of the SH3300.

Some high performance disk controllers supply a WCLK which is used to strobe data out of the controller during a write. Connecting this WCLK to the WCLK input of the SH3300 can provide timing advantages in the design, since the WCLK out to write data out timing relationships of the controller can be more precisely controlled, easing the timing constraints at the SH3300 inputs. In one-bit NRZ write mode, the SH3300 accepts NRZ data on NRZ<0>, clocked by WCLK. In two-bit NRZ write mode, the SH3300 accepts NRZ data in parallel on NRZ<1> and NRZ<0> clocked by WCLK

4.4 Transition Detector

Gain Control and Timing Recovery both require information about the locations of transition responses in the sample stream. Because the gain and timing loops cannot tolerate the delay of the sequence detector, a simple transition detector is provided. This transition detector is not normally used for data recovery, although its output may be selected in place of the sequence detector output by setting a control register bit. The peak qualification threshold, V, is also stored in a control register. Table 4.4-1 shows the conditions which must be satisfied to detect a transition during sample period n under various conditions, where:

Y_i is the signal sample for period i, and

V is the Qualification Threshold.

The qualification threshold should be set high, after which its appropriate value can be determined by calibration.

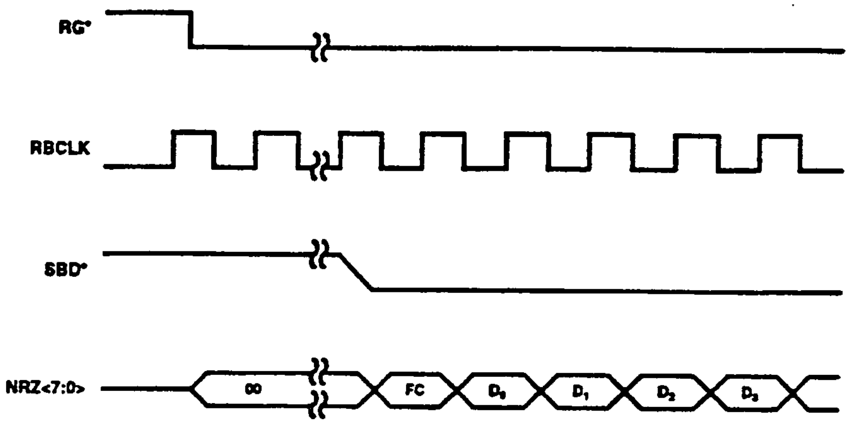
For an isolated transition response sample sequence of (a,b,1,c), the transition detector indicates a transition when it receives the fourth sample (c). The choice of either condition (3a) or (3b) during center-sampled acquisition is determined by a control register bit. Condition (3a) must be used with a '1010...' acquisition pattern; condition (3b) is intended for use with a '10001000...' acquisition pattern. Either may be used witha 100100..... acquisition pattern.

Table 4.4-1 Transition Detection Rules

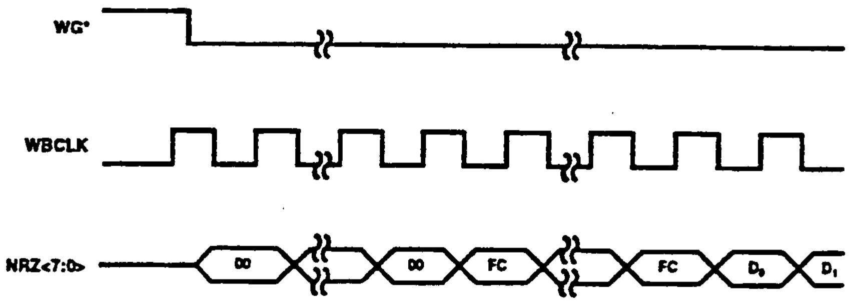
	Acquisition(+pulse)	Acquisition (- pulse)	Tracking (+ pulse)	Tracking (- pulse)
SAMPLING	$SGN(Y_{n-1}) = 1$	$SGN(Y_{n-1}) = -1$	$SGN(Y_{n-1}) = 1$	$SGN(Y_{n-1}) = -1$
Side sampled	1) $Y_{n-2} > Y_n$ 2) $Y_{n-1} > Y_{n-3}$ 3) $Y_{n-2} > Y_{n-4}$	1) $Y_{n-2} < Y_n$ 2) $Y_{n-1} < Y_{n-3}$ 3) $Y_{n-2} < Y_{n-4}$	1) $Y_{n-2} > Y_n$ 2) $Y_{n-1} > Y_{n-3}$ 3) $Y_{n-1} > V$	1) $Y_{n-2} < Y_n$ 2) $Y_{n-1} < Y_{n-3}$ 3) $Y_{n-1} < -V$
Center sampled	1) $Y_{n-1} > Y_n$ 2) $Y_{n-1} > Y_{n-2}$ 3a) $Y_{n-2} > Y_{n-3}$ 3b) $Y_{n-2} > Y_{n-4}$	1) $Y_{n-1} < Y_n$ 2) $Y_{n-1} < Y_{n-2}$ 3) $Y_{n-2} < Y_{n-3}$ 4) $Y_{n-2} < Y_{n-4}$	1) $Y_{n-1} > Y_n$ 2) $Y_{n-1} > Y_{n-2}$ 3) $Y_{n-1} > V$	1) $Y_{n1} < Y_n$ 2) $Y_{n-1} < Y_{n-2}$ 3) $Y_{n-1} < -V$

9.3.6 Operational Timings

9.3.6.1 Read Synchronization Operation (Eight Bit NRZ Interface)



9.3.6.2 Write Synchronization Operation (Eight Bit NRZ Interface)



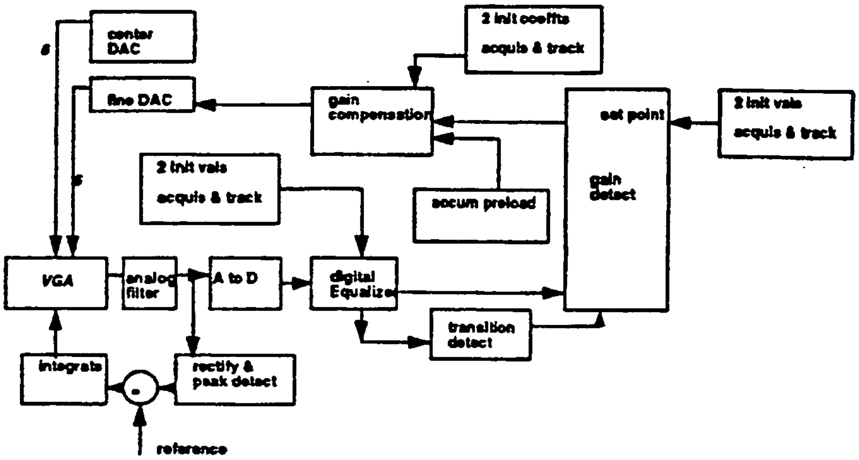


Figure 4.5.1. Gain Loop for data and servo

a, b, and c are Detector Levels 1, 2, and 3,

and T_i is 1 when a transition is detected during sample

period i and pulse shape compensation is enabled, 0 otherwise.

The nominal value of the Gain Set Point for side sampling is (1 + b); for center sampling, it is (1). The appropriate value of the Gain Set Point for normal operation is determined during calibration. Varying the Gain Set Point may be a suitable tool for retry strategies. For example, a larger Gain Set Point could be used to hold the gain artificially high, which might tend to compensate for drop-out errors (at some cost in signal-to-noise ratio). Note that the Gain Set Point affects the loop gain and gain margin of the Timing Recovery circuit.

Table 4.5-1 Gain Error

	Acquisition	Tracking
side sampled	$SGN(Y_{n-1})^*(Y_{n-1}+Y_{n-2})-g_a+(a+c)^*T_{n-2}$	$SGN(Y_{n-1})^*(Y_{n-1}+Y_{n-2})-g_t+(a+c).T_{n-2}$
center sampled	$SGN(Y_{n-1})^*Y_{n-1} - g_a + a.T_{n-2}$	$SGN(Y_{n-1})^*Y_{n-1} - g_t + a.T_{n-2}$

4.5.2 Gain Compensation

Compensation in the Gain Control loop for data reads is accomplished with a digital filter whose structure may be represented by Figure 4.5-1. The adder is designed to saturate rather than wrap on over/under flow.

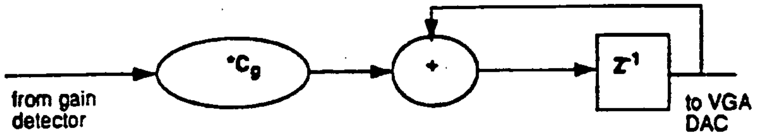


Figure 4.5-1 Gain Control Loop Compensation

The z-domain transfer function of this filter is

$$Fg(z) = \frac{c_g}{z-1}$$

c_g is the loop filter coefficient and z^{-1} represents a delay of two channel bit intervals ($2/f_S$) because this digital filter operates at one-half the channel sample rate. Coefficient c_g is independently programmable for acquisition and tracking and for servo and data.

A linear discrete-time approximation to the open-loop transfer function of the Gain Control loop is

$$Gg(z) = \frac{k_g \bullet k_a \bullet c_g}{z^n \bullet (z-1)}$$

where K_g is the gain of the gain detector (equal to one), K_a is the control gain of the VGA (equal to 0.25dB/ LSB, c_g is the gain of the compensation filter, and n is the number of ($2/f_S$) clock delays <TBD> in the loop. This model can be used to predict the acquisition settling time and the tracking bandwidth of the loop for a given filter coefficient c_g.

4.6 Timing Recovery

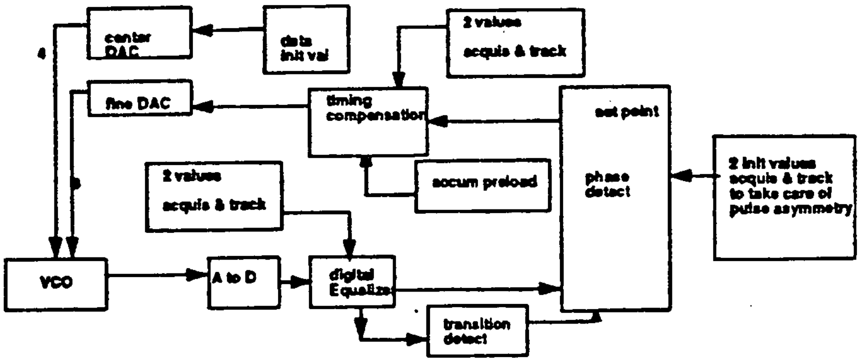
The Timing Recovery circuit controls the Read Clocks, which are synchronized to the read waveform. The Read Clocks are generated from a clock produced by a variable frequency oscillator (VFO) which is used to clock an analog-to-digital converter (ADC). A phase detector circuit digitally computes the phase error in the ADC sampling instants from the sample values. The sequence of measured phase errors is digitally filtered in the SH3300 to produce a frequency control signal which is fed back to the VFO. The digital value is converted via a DAC to an analog signal which acts to offset the VFO from its center frequency to establish the correct sampling frequency and phase. The phase detector, filter, VFO, and ADC together form a phase-locked-loop. The Timing Recovery circuit is designed so that the digital control will saturate instead of overflow or underflow.

The Timing Recovery circuit provides four modes: lock-to-reference, acquisition, tracking, and hold. In lock-to-reference mode, a digital phase/frequence detector is used to lock the read VFO to the frequency synthesizer clock, ensuring small frequency error when acquisition mode is entered.

In acquisition mode, which is automatically entered on the leading edge of Read Gate, the loop has been programmed to respond quickly in order to allow fast acquisition over the preamble. To further decrease acquisition time, a zero-phase start capability is included. Timing frequency and phase are assumed to have been acquired a programmable number of clocks after the leading edge of Read Gate. The Timing Recovery circuit then switches to a low-bandwidth tracking mode in which the loop has been programmed to respond more slowly in order to minimize timing errors over data due to noise. This change in response is accomplished by switching to another set of loop filter coefficients.

In hold mode, which is entered while Read Gate is deasserted, Timing Recovery action is suspended, allowing the digital control to "coast" at its current frequency e.g. over gaps.

Figure 4.6.1 Timing Recovery Loop



4.6.1 Phase Detector

The Phase Detector calculates the difference in phase between the desired and actual sampling instants. A timing update is made whenever a transition is detected. The timing update value is positive when the ADC is strobed late, indicating that the VCO frequency should be increased. The timing update value is negative when the ADC is strobed early, indicating that the VCO frequency should be decreased. Table 4.6-1 shows the equations used to calculate the phase error when a transition is detected during sample period n, where

SGN(x) is +1 for x GTE 0, -1 for x LT 0,

Y_i is the signal sample for period i,

t_a is the acquisition Timing Set Point,

t_t is the tracking Timing Set Point,

a, b, and c are Detector Levels 1, 2, and 3, and

T_i is 1 when transition is detected during sample period i and pulse shape compensation is enabled, 0 otherwise.

The Timing Set Point is used to accommodate pulse asymmetry. Table 4.6-2 shows the nominal values of t_a and t_t. The appropriate values of the Timing Set Point for normal operation are determined during calibration. Varying the values of the Timing Set Point may be a suitable tool for retry strategies.

Table 4.6-1 Phase Error

	Acquisition	Tracking
side sampled	$SGN(Y_{n-1})^*(Y_{n-2}- Y_{n-1})+t_a$	$SGN(Y_{n-1})^*(Y_{n-2} -Y_{n-1})+t_t + c^*T_{n-2}-a^*T_{n+2}$
center sampled	$SGN(Y_{n-1})^*(Y_{n-2}- Y_n) + t_a$	$SGN(Y_{n-1})^*(Y_{n-2} - Y_n)+t_t + c^*T_{n-2}-a^*T_{n-3}$

Table 4.6-2 Nominal Timing Set Point

	Acquisition pattern	Nominal t_a	Nominal t_t
side sampled	1010.....	1 - b + c - a	1 - b
	100100......	1 - b	
	10001000....	1 - b	
center sampled	1010.....	2 * (c - b)	c - b
	100100......	c - b + a	
	10001000....	c - b	

4.6.2 Timing Compensation

Compensation in the Timing Recovery loop is accomplished with a digital filter whose structure may be represented by Figure 4.6-1.

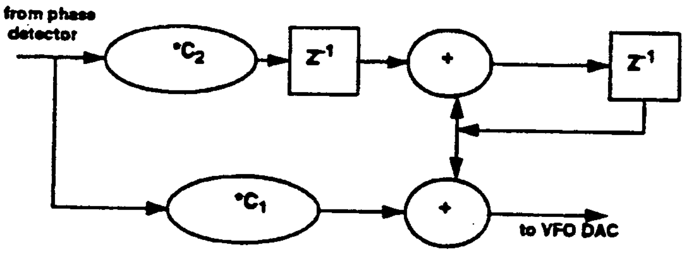


Figure 4.6-1 Timing Recovery Loop Compensation

The z-domain transfer function of this filter is

$$F_t(z) = \frac{c_1 \bullet z \bullet (z-1) + c_2}{z \bullet (z-1)}$$

Where z^{-1} represents a delay of two channel bit intervals ($2/f_S$) because this digital filter operates at one-half the channel sample rate. A linear discrete-time approximation to the open-loop transfer function of the Timing Recovery loop is

$$G_t(z) = \frac{k_p \cdot k_o \cdot (c1 \bullet z^2 - c1 \bullet z + c_2)}{z^{n+1} \bullet (z-1)^2}$$

where K_p is the gain of the phase detector (1 to 2 depending on pulse shape), K_o is the control gain of the VCO (effectively 1/2 for this model), and n is the number of ($2/f_S$) clock delays <TBD> in the loop. This model can be used to predict the acquisition settling time and the tracking bandwidth of the loop for a given pair of filter coefficients c_1 and c_2. Given detector levels a, b, and c, the gain of the phase detector may be computed as $K_p = 1 + b - 3^*(a + c)/2$ for side sampling. For center sampling, the gain of the phase detector is $K_p = 2 - 2^*a/3 - (b + c)/2$.

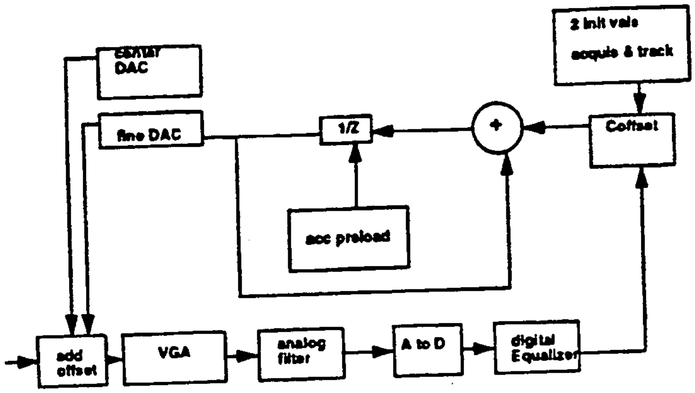
The coefficients c1 and c2 can each have two values, for acquisition/tracking. Additionally, the filter accumulator can be initialized at the start of acquisition with either an initial coefft. or the previously saved accumulator value at the end of the last data field.

4.7 Offset Loop

The offset loop in SH3300 is a closed loop system that acts at the VGA to minimize offsets at the output of the A/D. The offset is controlled by a coarse DAC set statically through registers.. The closed loop control is through a fine DAC that is controlled by averaging the digital samples to ensure that positive and negative transitions are forced to be equal. Any difference in positive and negative transition amplitudes will be interpreted as an offset. The loop may be disabled by setting the filter parameter Coffset to zero. The loop also serves to lessen the effect of 1/f noise in the VGA.

Figure 4.7-1 shows the structure of the offset loop

The coefficient Coffset can be independently set for acquisition/tracking. Additionally the filter accumulator can be initialized at the start of acquisition with either an initial coefft. or the previously saved accumulator value at the end of the last data field.



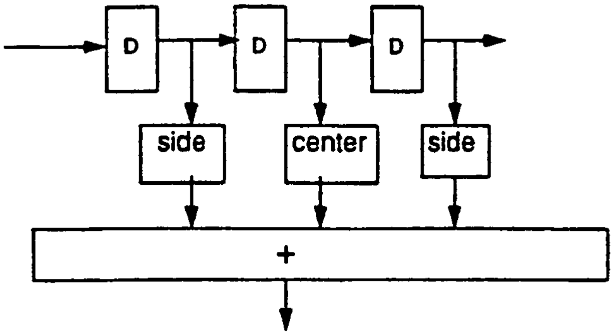
4.8 Digital Equalizer

The digital equalizer can modify the equalization done in the analog filter and, along with it, provides support for changing equalization needs from head to head and zone to zone.

The equalizer parameters are loaded by the microcontroller at initialization and during seeks. Using the Channel Quality circuit, a procedure is provided whereby the microcontroller can adapt the equalizer parameters.

The intent of this feature is to allow the microcontroller to find the best parameter sets (those which produce the minimum error rate) for each disk drive, head, and zone. Since parameter optimization is a slow process, it is done only during manufacturing test and/or during idle times. During servo processing, under control of the SVEN input, the filter coefficients will become those previously programmed in the servo coefft. registers

Figure 4.8.1 shows the structure of the Digital Filter.



Furthermore, the digital filter outputs go into a programmable gain stage (re-scaler) which offers programmable gain values of 0.5, 1 and 2. Again these values can be set independently for data and servo. Each of the side coefficients can be independently set.

Table 4.8.1 Digital filter coefficients

side coefficients	center tap coefficien
-6/16 to +6/16 in 1/16 steps	0.5
	0.75
	1.00
	1.25

Figure 4.8-2 shows the Digital Filter amplitude response vs. frequency

Figure 4.8.2 Digital Filter Response

4.9 Spectral Smoothing Filter

The Spectral Smoothing filter is designed to lessen pole-tip effects when using thin-film heads, or secondary gap effects when using MIG heads. The filter employs a center tap and two groups of side coefficients separated by programmable delays. The entire filter or just its precursor-correcting portion can be disabled. Fig 4.7.2 shows the Spectral Smoothing Filter. The delays n1 and n2 can be independently set from 2 to 23 delay instances.

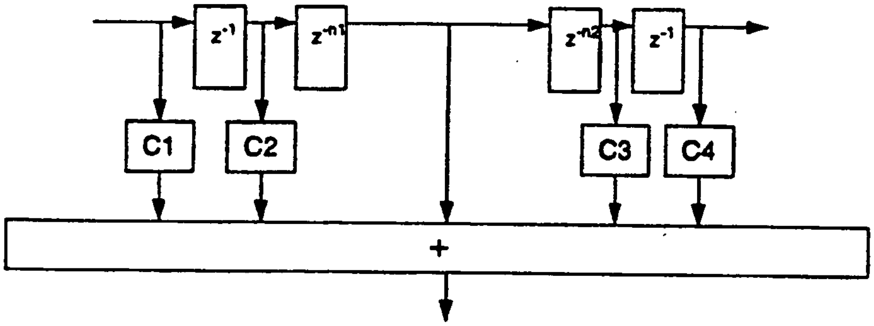


Figure 4.9.1 SFilter

The coefficients C1 and C2 have the same sign, + or -.

The coefficients C4 and C5 have the same sign + or -

each coefficient can be independently programmed 0, 1/32, 1/16

4.10 Analog Equalization

The analog equalizer is a 7th order equi-ripple filter with 0.05 degree phase ripple. The filter has programmable cut-off and boost. The cut-off frequency is determined by two parameters. First, an external crystal reference that will typically be at the inside diameter NRZ rate, and secondly, by an 8-bit register that programs the unboosted cut-off frequency of the filter elative to the crystal frequency. The crystal frequency (fc) may be set between 10MHz to 30MHz, and the cut off may be varied in two ranges. The first range is from 3 to 15 MHz and the second range from 6 to 30 MHzc. The high frequency boost capability is a maximum of 12dB, above the DC amplitude value. This boost can be varied by programming a 4-bit register.

Figure 4.10.1 Analog Filter response

4.11 Sequence Detector

Partial response is a communications and recording technique used to recover, as much as is practical, the loss associated with inter-symbol interference (ISI). Partial response models for magnetic recording channels are typically defined by polynomials of the form

$$(1 - D)*(1 + D)^N$$

where D represents a unit delay. The $(1 - D)$ factor can be associated with differentiation by the transducer and the $(1 + D)^N$ factor defines the intersymbol interference. N would be zero for a recording channel without intersymbol interference. N is one for Partial Response Class IV (PR4) and two for Extended Partial Response Class IV (EPR4). Higher orders of $(1 + D)$ are also possible with SH3300 and are tabulated below. . The expansion of $(1 + D)^N$ defines the sample points of an isolated pulse. Note that when N is even there is a sample at the center of the isolated pulse and that when N is odd all samples are on the sides of the isolated pulse.

	PARTIAL RESPONSE		NORMALIZED SAMPLE
N	POLYNOMIAL	NAME	VALUES
0	$(1 - D)$	Dicode	0 1 0
1	$(1 - D)*(1 + D)$	PR4	0 1 1 0
2	$(1 - D)*(1 + D)^2$	EPR4	0 1/2 1 1/2 0
3	$(1 - D)*(1 + D)^3$	EEPR4	0 1/3 1 1 1/3 0
	$(1-D)*(a+bd+D^2+cD^3)$	SofTarget	0 a b 1 c 0

We can take a more general view of the polynomial by varying coefficients. For example, replace the side coefficients of [1/2 1 1/2] with K to obtain [K 1 K]. As K varies from 0 to 1/2, the response changes from that associated with the polynomial $(1 - D)$ to that associated with $(1 - D)*(1 + D)^2$. The corresponding time and frequency responses are shown in Figures 4.8-1 and 4.8-2, respectively. Similarly, replace the side coefficients of [1/3 1 1/3] with K to obtain [K 1 1 K]. As K varies from 0 to 1/3, the response changes from that associated with the polynomial $(1 - D)*(1 + D)$ to that associated with $(1 - D)*(1 + D)^3$. The corresponding time and frequency responses are shown in Figures 4.8-3 and 4.8-4, respectively. We need not equalize strictly to a target defined by $(1 - D)*(1 + D)^N$; a pulse shape in between may be a better match to a particular channel.

At significant user densities, the signal-to-noise ratio required for the sequence detector to produce a given error rate is generally several dB lower than that required for digital peak detection. This advantage is generally greater for negatively-correlated noise, which is enhanced when high-frequency boost is employed to slim the pulse shape, and generally less for positively-correlated noise.

Figure 4.11-1

Figure 4.11-2

Figure 4.11-3

Figure 4.11-4

Figure 4.11-5 shows the Sequence Detector, which reconstructs the channel-bit stream from the analog- and digitally-equalized samples not on a bit-by-bit basis from each sample in turn but rather with regard to both the current sample and the surrounding sequence of samples according to a decision rule which produces near-maximum-likelihood detection. The detection algorithm determines the best sequence of RLL 1's and 0's corresponding to the pattern of samples.

Figure 4.11-5 Sequence Detector

The Sequence Detector can be programmed to operate on any channel response which can be well represented by sequences of the form [a b 1 c].

center sampled pulses nominally have

$$a < b = c < 1$$

side sampled pulses nominally have

$$a < 1, b = 1, c < 1$$

Pulse asymmetry is accommodated by specifying:

for center sampling:

$$a \neq 0$$

and

$$b \neq c$$

for side sampling

$$a \neq c$$

and / or

$$b \neq 1$$

Using the Channel Quality circuit, a procedure is provided whereby the microcontroller may adapt the detector sample levels. The intent of this feature is to allow the microcontroller to find the best detector sample level sets (those which produce the minimum error rate) for each disk drive, head, and zone. Since parameter optimization is a slow process, it is done only during manufacturing test and/or during idle times.

4.12 Synchronization Mark Detection

The error-tolerant synchronization strategy implemented in the SH3300 is designed to achieve a level of error tolerance for the synchronization function equal to that achieved for the data field by the combination of the Sequence Detector in the SH3300 and the error correction code implemented in the Disk Controller. This strategy employs an error-tolerant Synchronization Mark pattern which is selected for minimum cross-correlation with the preamble and for minimum auto-correlation. The length of the error-tolerant Synchronization Mark and the number of four-channel-bit groups which must be detected are programmable.

When Read Gate is asserted, the SH3300 begins to emit all zeros on the NRZ Data interface. When the SH3300 enters tracking mode as specified by the Acquisition Length, it begins to attempt to detect the Synchronization Mark Threshold. When the Synchronization Mark Threshold is met, the SH3300 emits a hex '0D' byte over the NRZ Data interface ,followed by decoded data bits after the synch field.. Windowing of the detection of the Synchronization Mark is perfomed in the Disk Controller using the known length of the preamble and Synchronization Mark.

The Synchronization Mark Recovery procedure may be used to recover data when a severe defect has destroyed the entire Synchronization Mark. To use this mode, the Disk Controller asserts the Sequencer Output signal before Read Gate. The SH3300 first goes through a normal gain and timing acquisition procedure while counting channel bits. The Synchronization Mark is assumed to have been detected when the count matches the Synchronization Mark Recovery Count. By varying the Synchronization Mark Recovery Count, the microcontroller can vary the assumed starting point of a header or data area until the correct starting point is tried, whereupon the sector will be recovered if there is no other error beyond the capability of the error correction code in the Disk Controller. A conservative strategy to limit the probability of miscorrection would be to limit error correction to a single burst when the Synchronization Mark Recovery procedure is used.

4.12.1 Synchronization Mark Pattern

The RLL-bit pattern of the error-tolerant Synchronization Mark is programmed by the user in the Synchronization Mark Pattern registers. Cirrus recommends the patterns shown in Table 4.9-1.

Table 4.9-1 Recommended Synchronization Mark Patterns

Acquisition pattern	Synchronization Mark Pattern
1010.....	10010010 10001001 00010001 01010010 10000100 101010101
100100......	TBD
10001000...	TBD
.	

When the Synchronization Mark Length is programmed for one, two, or three NRZ bytes, the first twelve, twenty-four, or thirty-six bits, respectively, of the pattern are used. When the Synchronization Mark Length is programmed for one, two, three, or four bytes, the Synchronization Mark Threshold should be programmed to require error-free detection of three, four, six, or nine four-channel-bit groups, respectively, in normal operation.

4.13 RLL Encoder/Decoder

During Write operations, NRZ data bits are input from the Disk Controller and encoded according to the rules of a rate 2/3 RLL (1,7) code. The encoded RLL channel bits are then transmitted after write precompensation to the write head.

During Read operations, channel bits are output from the Sequence Detector and decoded according to the rules of the RLL (1,7) code. The decoded NRZ bits are then output to the Disk Controller.

The RLL (1,7) code can be encoded using one data-word look-ahead and one channel-bit look-back, and can be decoded using one channel-word look-ahead. Table 4.10-1 shows the encode/decode mapping; x means the complement of the preceding channel bit. The maximum length of an NRZ error burst caused by a single RLL drop-out, drop-in, or bit-shift error is five NRZ bits.

Table 4.10-1 RLL (1,7) Encode/Decode Mapping

NRZ	RLL
00	010
10	X01
11	X00
0100	010001
0101	010000
0110	X00001
0111	x00000

Note that '1010...', '100100...', and '10001000...' acquisition patterns are encoded from repeated hex '88', 'FF', and 'D3' bytes, respectively.

4.13.1 Data Randomizer

The RLL Encoder/Decoder also includes a Data Randomizer which processes unencoded user data to insure that channel bit patterns with worst-case pattern sensitivity occur no more frequently than would be expected from random user data. The Data Randomizer employs two linear feedback shift registers. One generates a 63-bit sequence which is EXCLUSIVE-OR-ed against the MSB of each pair of data bits. The other generates a 127-bit sequence which is EXCLUSIVE-OR-ed against the LSB of each pair of data bits. The Data Randomizer does not affect error propagation. When the Data Randomizer is enabled, the probability of encountering any specific pattern of length n channel bits at a randomly selected location within encoded data is approximately $1/2^n$.

4.14 Erasure Pointers

Normally, NRZ data bits are transmitted over the NRZ bus. When erasure pointer generation is activated, erasure pointer bits are transmitted instead. Erasure pointer bits are generated by the detection of maximum run-length constraint violation in the RLL Decoder. An erasure pointer bit in a given byte could indicate the existence of an error in that byte, a previous byte, or a following byte.

4.15 Channel Quality

The nature and quality of the channel comprising the head/disk interface, preamp, tunable active filter, ADC, and digital equalizer can be measured with the Channel Quality circuit, which is provided so that the variable channel parameters (e.g. analog and digital equalization parameters, gain and timing set points, gain control and timing recovery coefficients, and detector levels) may be adjusted to provide the lowest possible error rate. The Channel Quality circuit provides two modes of operation: Pulse Measurement and Metric Measurement. Each mode requires that the sequence that was written to the medium be known as the sequence is then read. This is accomplished by generating the written sequence entirely within the SH3300 Data Randomizer. Then on read, the same sequence can be generated and used in controlling the Channel Quality circuit. Either a pseudo-random or a repeating sequence can be generated by leaving feedback within the Data Randomizer enabled or disabled, respectively.

In the Pulse Measurement Mode, as the known sequence is read the Channel Quality circuit measures the cumulative squared differences of those samples corresponding to one of the combinations of the nominal sample points (e.g. a, b, 1, c, c-a, 1-b, b-c). The source used for pulse measurement is selectable between the sampled data output of the A/D , the output of the Digital Filter, and the output of the spectral smoothing filter. The cumulative squared differences can be used to determine the pulse shape produced by a given set of channel parameters and to adapt the channel parameters to produce a desired pulse shape.

In the Metric Measurement Mode, as the known sequence is read, the Channel Quality circuit measures the cumulative metric of the final state of the sequence. The source used for metric measurement is selectable between the sampled data output of the A/D, the output of the Digital Filter, or the output of the spectral smoothing filter. The cumulative metric can be used to adapt the channel parameters and to predict the error rate.

4.16 Power Management

The SH3300 has three basic power down modes. In idle mode, all elements are switched off. The SH3300 would typically be in idle mode when the disk has been spun down. As the disk spins up SH3300 should be switched out of idle mode.

In track following mode, some elements are still powered down, but other elements are powered up to process servo bursts. The SVEN and SVRD signals are used to determine the arrival of a servo field.

Normal mode is used for full reading or writing.

Table 4.16.1 shows the different elements of SH3300 and indicates if the element is off, on or under the control of an external signal.

Table 4.16.1 SH3300 Power Down Modes

circuit	idle	track follow	normal	note
VGA	off	SVEN	SVEN + RG	
analog filter	off	SVEN	SVEN + RG	
A/D	off	SVRD	SVRD + RG	
VCO	off	SVEN	on	
SYNTH	off	off	on	
zero phase restart	off	acquisition	acquisition	
write-precomp	off	off	WG	
servo rectifiers	off	SVEN	SVEN	
track & holds	off	on	on	
digital filter	off	SVRD &configured	(SVRD + RG) & configured	1
Spectral Smooth filter	off	off	TRK & configured	2
sequence detector	off	off	TRK	
servo bit recovery	off	track	SVEN & TRACK	
transition detector	off	SVRD	RG + SVRD	
gain loop filter/detect	off	SVRD	RG + SVRD	
timing loop filt/detect	off	SVRD	RG + SVRD	
encoder	off	off	WG	
decoder	off	off	TRK	
sync detect	off	off	TRACK	

circuit	idle	track follow	normal	note
FF detect	off	off	WG	
VGA Common Mode	off	SVEN	WG	3
VREF	off	SVEN	ON	

Notes:

1) Configured means the particular element has been configured active by muxing

2) TRK means that the SH3300 is in tracking (versus acquisition) mode

3) After WG goes inactive, VGA remains active for a specified time to allow read amp to recover (squleich)

4.17 Error Budget

When designing a recording channel with the SH3300 , an error budget must be established. An error budget might include the following:

- Media Noise
- Non-linear Write Transition Shift
- Partial-Erasure Amplitude Non-linearity
- Overwrite Noise
- Offtrack Noise
- Preamp Electronics Noise
- Amplifier/filter Electronics Noise
- ADC quantization Error
- Misequalization Noise
- Gain Error
- Timing Error
- Margin

4.17.1 Media Noise

4.17.2 Non-linear Write Transition Shift

4.17.3 Partial-Erasure Amplitude Non-linearity

circuit	idle	track follow	normal	note
FF detect	off	off	WG	
VGA Common Mode	off	SVEN	WG	3
VREF	off	SVEN	ON	

Notes:

1) Configured means the particular element has been configured active by muxing

2) TRK means that the SH3300 is in tracking (versus acquisition) mode

3) After WG goes inactive, VGA remains active for a specified time to allow read amp to recover (squelch

4.17 Error Budget

When designing a recording channel with the SH3300 , an error budget must be established. An erro budget might include the following:

- Media Noise
- Non-linear Write Transition Shift
- Partial-Erasure Amplitude Non-linearity
- Overwrite Noise
- Offtrack Noise
- Preamp Electronics Noise
- Amplifier/filter Electronics Noise
- ADC quantization Error
- Misequalization Noise
- Gain Error
- Timing Error
- Margin

4.17.1 Media Noise

4.17.2 Non-linear Write Transition Shift

4.17.3 Partial-Erasure Amplitude Non-linearity

4.17.4 Overwrite Noise

4.17.5 Offtrack Noise

4.17.6 Preamp Electronics Noise

4.17.7 Amplifier/Filter Electronics Noise

4.17.8 ADC Quantization Error

4.17.9 Misequalization Noise

4.17.10 Gain Error

4.17.11 Timing Error

4.17.12 Margin

4.18 Servo Subsystem

The servo subsystem in SH3300 is designed to provide a standard interface to the servo decoder and servo A to D system. The SH3300 servo system is responsible for peak detecting the servo preamble, ID field and track number, and passing this peak detected digital data out to the servo decoder.

The SH3300 is switched out of data mode and into servo mode by the assertion and strobing of the SVEN command on SVENC<3:0>. This command should be derived from a timer that estimates the occurence of the current servo burst from the time of the previous servo burst. On the assertion of this encode, the analog filter will switch to the appropriate servo coefficients. Likewise, VGA control will be switched from data to servo.

The servo preamble is acquired by the servo decoder, at which point the servo decoder should strobe in the SVRD command. This assertion of the SVRD command replaces those coefficients appropriate to the acquisition phase of the servo burts with the set of cofficients appropriate to the servo tracking phase.

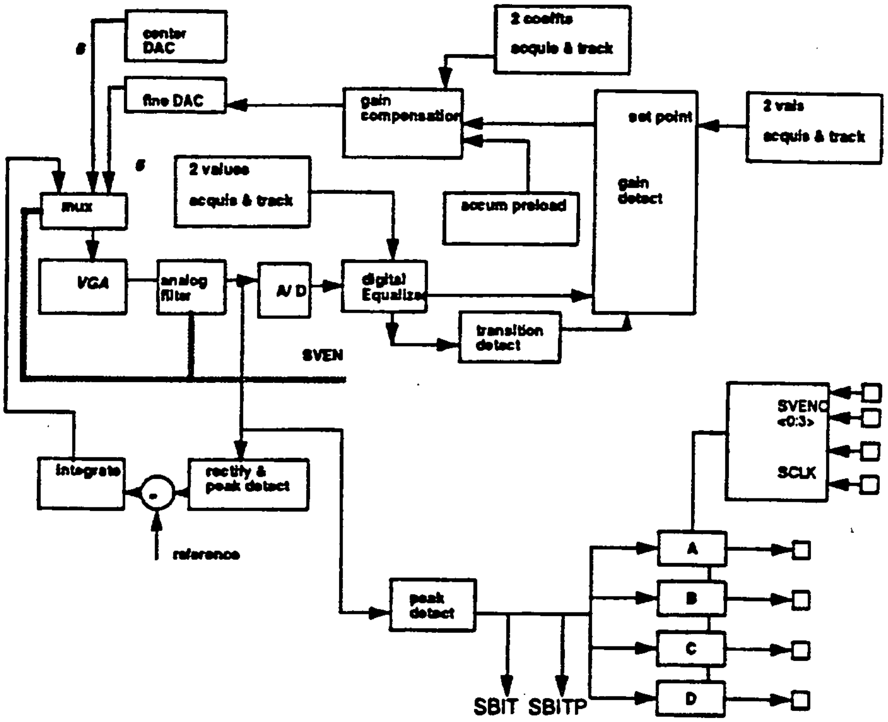
Figure 4.18.1 shows the servo data recovery block diagram

.The servo burst information is peak detected in the analog domain, and output can be captured by 4 sample and hold circuits which drive 4 output pins. The BSAMP command controls the sampling of the peak detected output, and additionally, a DUMP command is provided to allow the user flexibility on when the sample and hold circuits will be reset. In this manner, different servo burst schemes can be accommodated.

The peak detect block for the analog servo burst is fed from the output of the analog filter. Four sample and hold circuits are used to capture the output of the peak detect, under the control of different BSAMP arguments. These commands can be strobed as appropriate from the external servo timing logic.

SVENC<3:0>	servo function
0000	switch into acquisiton mode
0001	switch into servo mode
0010	switch into tracking mode
0011	sample A burst
0100	Sample B burst
0101	sample C burst
	sample D burst
	Enable servo burst dump
	switch off all sample and holds
	hold AGC
	Switch out of servo mode (into data mode)
	power down servo block
	power up servo block

Fig 4.18.1 Servo Detection



4.19 Calibration

The sampled-amplitude channel technology employed in the SH3300 makes it possible to implement adaptive channels. These channels are calibrated in manufacturing and/or during idle times and therefore accommodate greater variation in head/media parameters.

In a SH3300 implementation, many equalization and detection parameters are under user control. These include:

- VGA DC Offset
- Tunable Filter Cutoff
- Tunable Filter Boost
- Gain Set Point
- Timing Set Point
- Gain Control Loop Coefficients
- Timing Recovery Loop Coefficients
- Digital Equalizer Parameters
- Center/Side Sampling
- Detector Levels

There are many possible tradeoffs; the user decides which parameters are fixed during development, which are to be adapted during manufacturing, which are to be re-adapted during idle times, and which are to be varied during retries. Note that a different set of parameters, including the choice of center sampling vs. side sampling, may be used for each head and zone on a given drive.

Figure 4.4-1 illustrates center sampling and side sampling for an isolated pulse. The a, b, and c labels refer to programmable Detector Levels 1, 2, and 3, respectively.

Figure 4.4-1.

4.5 Gain Control

The differential variable gain amplifier (VGA) in the SH3300 amplifies the differential input signal from the read preamp to a consistent level which is fed to the tunable active filter. For data read operation, the VGA gain is controlled by a digital to analog converter (DAC) with coarse gain statically controlled by a register and fine gain dynamically controlled by the Gain Control circuit of the SH3300 . The Gain Control circuit is designed for symmetrical attack and decay, and is also designed to ensure that the digital gain control will saturate and not overflow or underflow. Furthermore, the coding of the DAC control is thermometer coding, ensuring that a samil change in value cannot lead to DAC glitches in the VGA circuit.

During servo read operation, the VGA is controlled through an analog gain loop that bypasses the DAC control of the VGA.

The Gain Control circuit provides three modes: acquisition, tracking, and hold. In acquisition mode, which is automatically entered on the leading edge of Read Gate, the loop has been programmed to respond quickly in order to allow fast acquisition.

Proper gain is assumed to have been established a programmable number of clocks after the leading edge of Read Gate. The Gain Control circuit then switches to a lower-bandwidth tracking mode in which the loop has been programmed to respond more slowly in order to minimize gain errors over data due to noise. This change in response is accomplished by switching to another set of loop filter coefficients.

In hold mode, which is entered while Read Gate is deasserted, Gain Control action is suspended, allowing the VGAC bus to "coast" at its current gain level, e.g. over gaps. Gain does not "droop" in hold mode due to the digital nature of the gain control loop.

4.5.1 Gain Detector

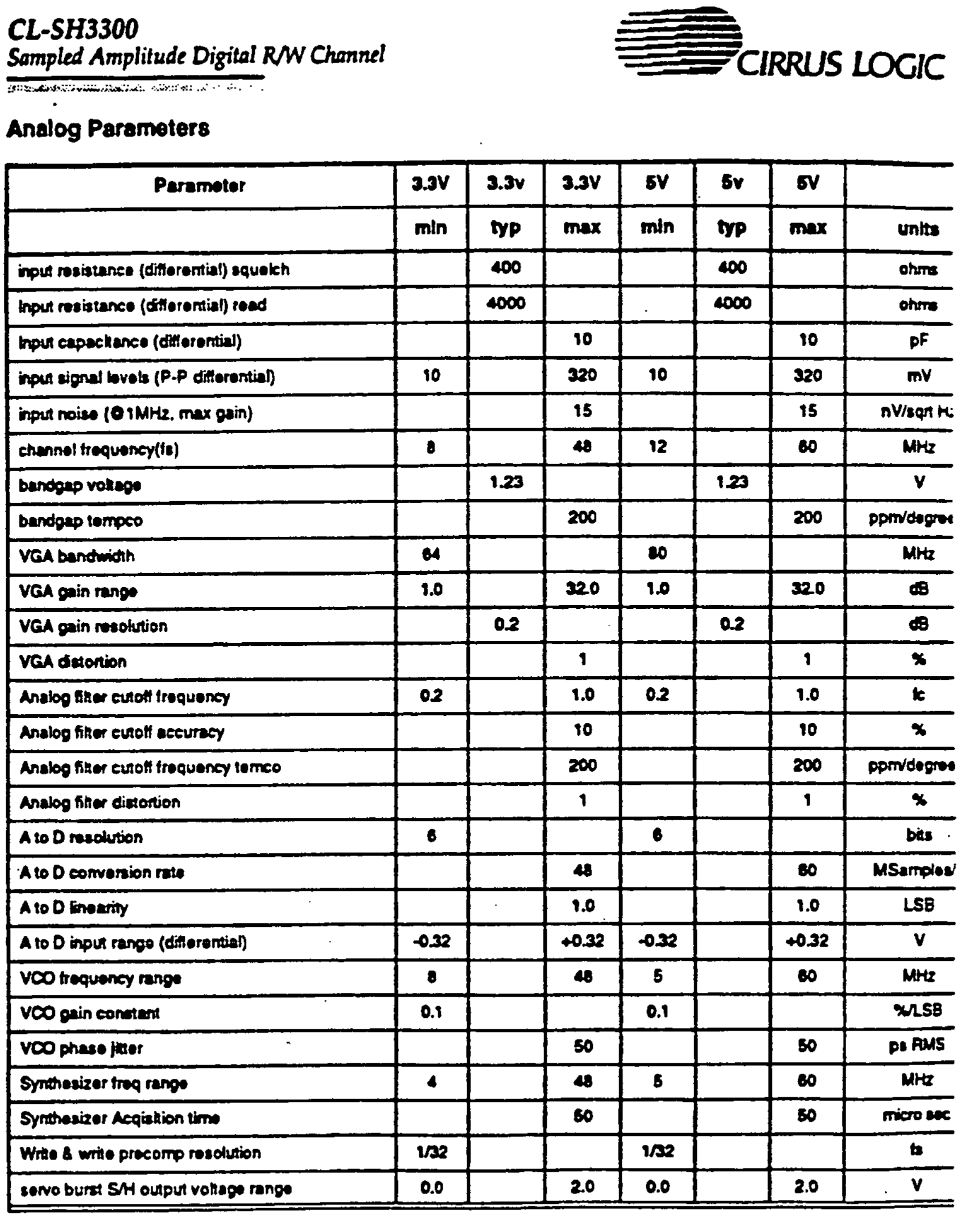
The Gain Detector calculates the difference in amplitude between the desired and actual signal levels. A gain update is made whenever a transition is detected. The gain update is negative when the gain is too high and positive when the gain is too low. Table 4.5-1 shows how the gain error is calculated when a transition is detected during sample period n, where

SGN(x) is +1 for x >= 0, -1 for x < 0,

Y_i is the signal sample for period i,

g_a is the acquisition Gain Set Point value,

g_t is the tracking Gain Set Point value,



CL-SH3300
Sampled Amplitude Digital R/W Channel

CIRRUS LOGIC

Analog Parameters

Parameter	3.3V	3.3v	3.3V	5V	5v	5V	
	min	typ	max	min	typ	max	units
input resistance (differential) squelch		400			400		ohms
input resistance (differential) read		4000			4000		ohms
input capacitance (differential)			10			10	pF
input signal levels (P-P differential)	10		320	10		320	mV
input noise (@1MHz, max gain)			15			15	nV/sqrt H
channel frequency(fs)	8		48	12		60	MHz
bandgap voltage		1.23			1.23		V
bandgap tempco			200			200	ppm/degree
VGA bandwidth	64			80			MHz
VGA gain range	1.0		32.0	1.0		32.0	dB
VGA gain resolution		0.2			0.2		dB
VGA distortion			1			1	%
Analog filter cutoff frequency	0.2		1.0	0.2		1.0	fc
Analog filter cutoff accuracy			10			10	%
Analog filter cutoff frequency temco			200			200	ppm/degree
Analog filter distortion			1			1	%
A to D resolution	6			6			bits
A to D conversion rate			48			60	MSamples/
A to D linearity			1.0			1.0	LSB
A to D input range (differential)	-0.32		+0.32	-0.32		+0.32	V
VCO frequency range	8		48	5		60	MHz
VCO gain constant	0.1			0.1			%/LSB
VCO phase jitter			50			50	ps RMS
Synthesizer freq range	4		48	5		60	MHz
Synthesizer Acqisition time			50			50	micro sec
Write & write precomp resolution	1/32			1/32			fs
servo burst S/H output voltage range	0.0		2.0	0.0		2.0	V

5.3 AC Characteristics

Unless otherwise specified, the following timings assume that all outputs will drive one Schottky TTL load in parallel with 50 pF and all inputs are at CMOS levels. The Min and Max timings conform to the operating ranges of power supply voltage (+/- 10 percent) and ambient temperature (0 to 70 ° C). All clocks are 60/40 percent maximum/minimum duty cycle. Rise and fall times should not exceed 3 ns.

5.3.1 Reset Assertion Timing Parameters

Symbol	Parameter	Minimum	Maximum	Units
Trpwl	RST* pulse width low	500		ns

5.3.2 Microcontroller Interface Timing Parameters

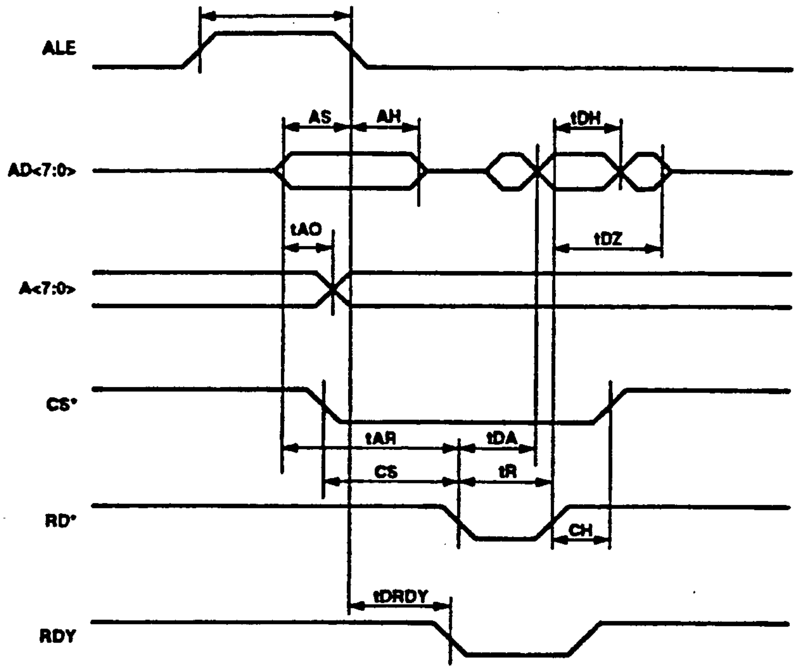
Symbol	Parameter	Minimum	Maximum	Units
Ta	ALE width	15		ns
Taw	Address valid to WR* _ or DS _	20		ns
Tar	Address valid to RD* _ or DS _	15		ns
Tao	AD<7:0> valid to A<7:0>		20	ns
	(Multiplexed mode)			
Tah	RD* _, WR* _ or DS _ to A<7:0> invalid	5		ns
	(Nonmultiplexed Mode)			
Tw	WR* width	60ns		
Tr	RD* width	60ns		
As	Address valid to ALE _	5		ns
Ah	ALE _ to address invalid1	0		ns
Cs	CS* _ to RD* _, WR* _, or DS _	0		ns
Ch	RD* _, WR* _ or DS _ to CS* _	5ns		
Wds	Write data valid to WR* _ or DS _	20		ns
Wdh	WR* _ or DS _ to Write data invalid	10		ns
Tda	RD* _ or DS _ to Read data valid		50	ns
Tdh	RD* _ or DS _ to Read data invalid	10		ns
Tdz	RD* _ or DS _ to Read data undriven		15	ns
Tds	DS width	60		ns
Tdth	DS _ to DTACK* _		20	ns
Tdrdy	ALE _ to RDY _		30	ns
Tsrw	R_W* valid to DS _	20		ns
Thrw	DS _ to R_W* invalid	20		ns

Note: _ indicates rising edge, _ indicates falling edge.

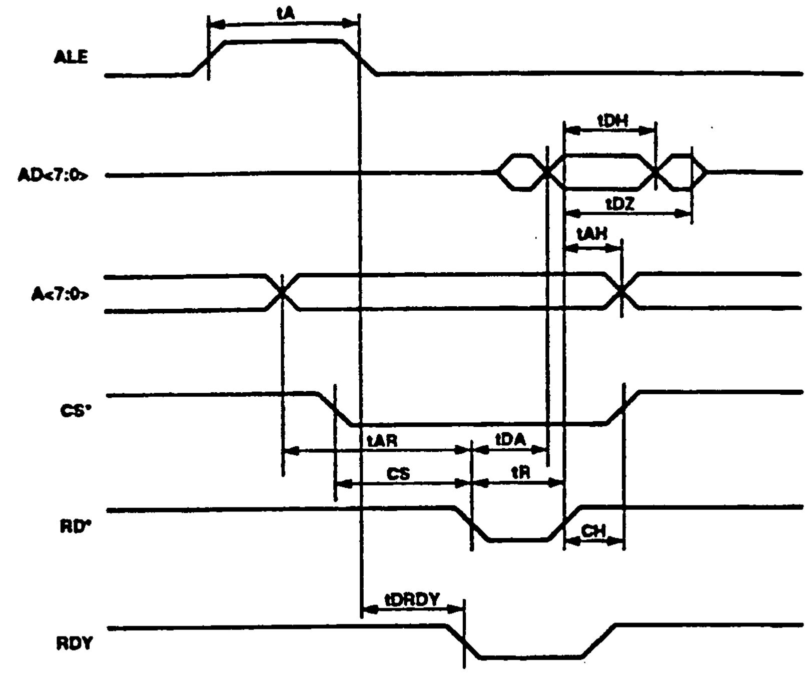
When I/MC* is high, the Intel bus control interface is selected. The timing diagrams in Sections 6.3.2.1 - 6.3.2.4 depict register read and write operations with this interface selected.

When I/MC* is low, the Motorola bus control interface is selected. The timing diagrams in Sections 6.3.2.5 - 6.3.2.8 depict register read and write operations with this interface selected.

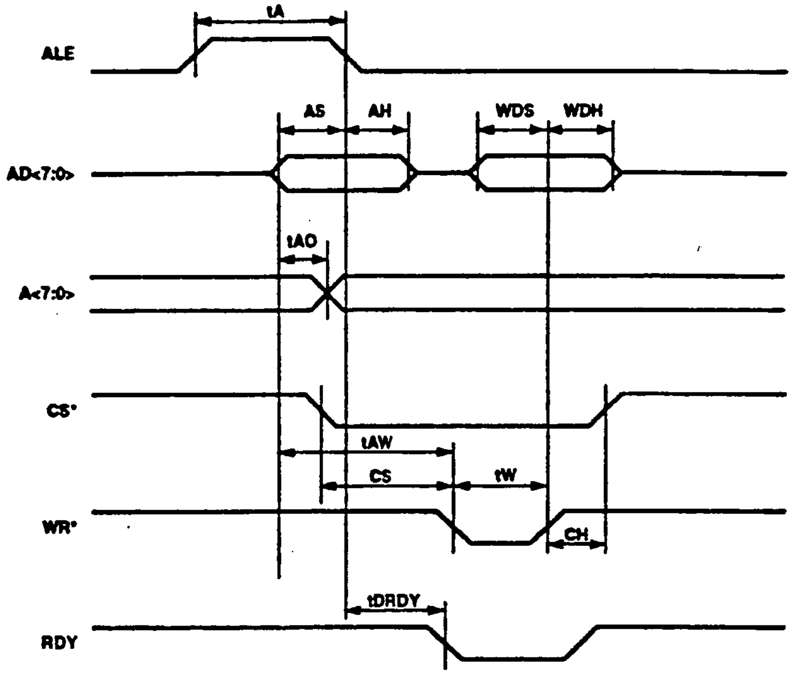
5.3.2.1 Register Read Operation in Intel Multiplexed Mode



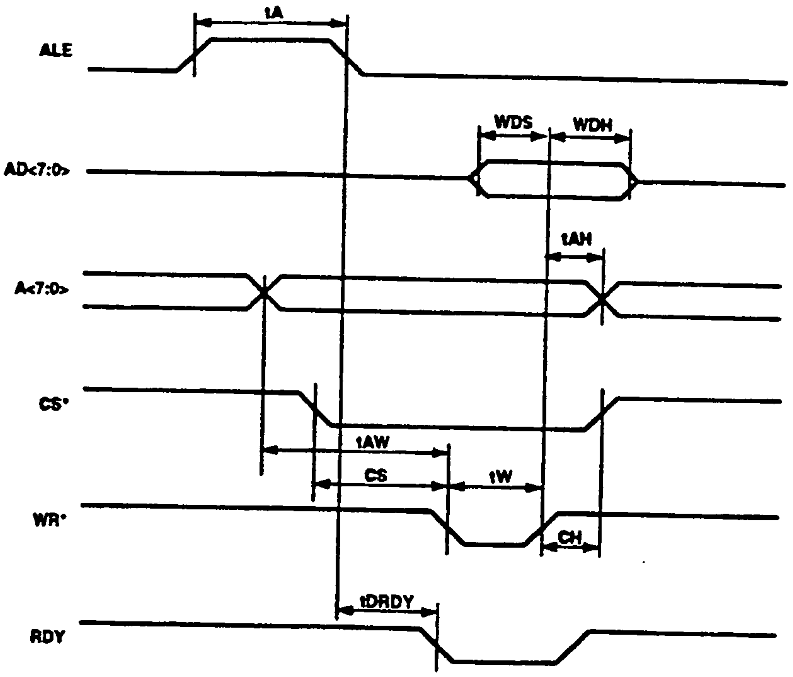
5.3.2.2 Register Read Operation in Intel Nonmultiplexed Mode



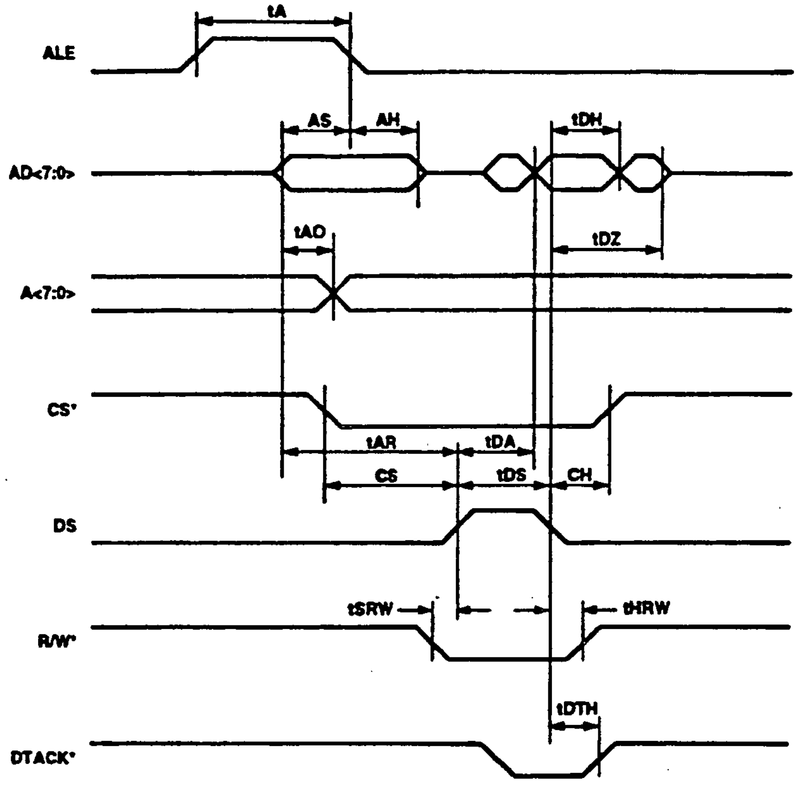
5.3.2.3 Register Write Operation in Intel Multiplexed Mode



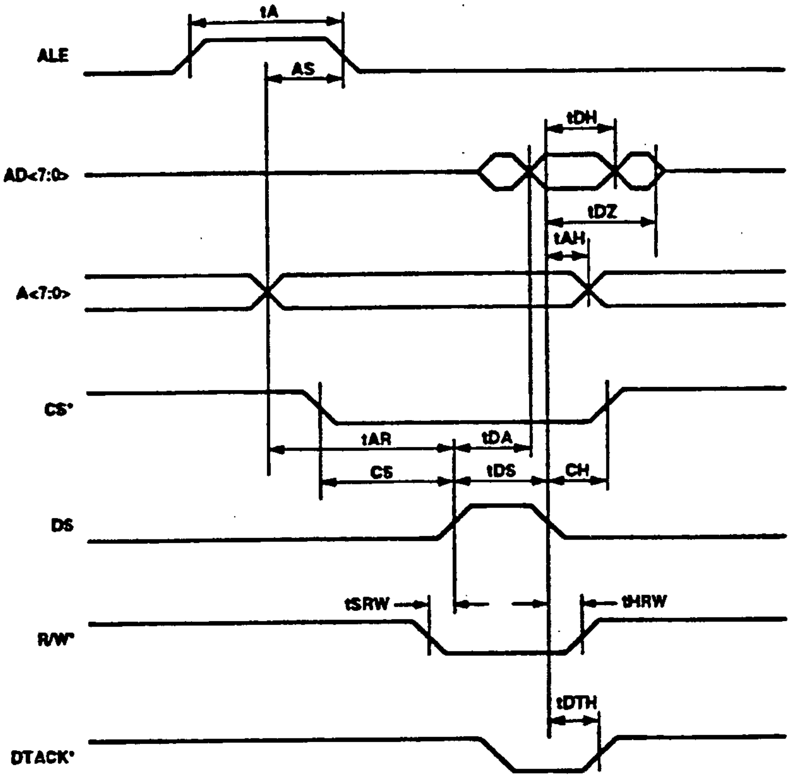
5.3.2.4 Register Write Operation in Intel Nonmultiplexed Mode



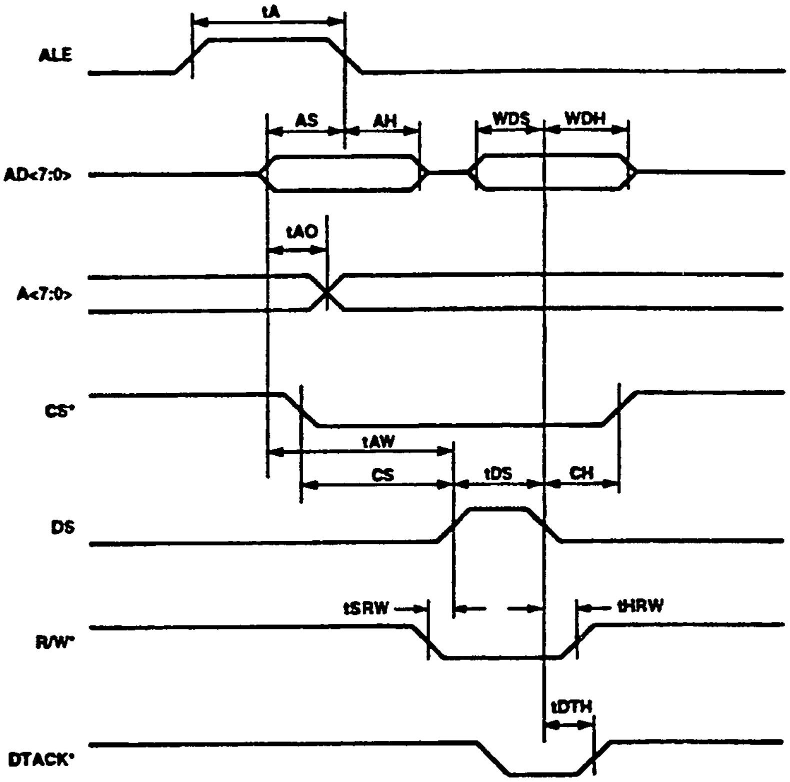
5.3.2.5 Register Read Operation in Motorola Multiplexed Mode



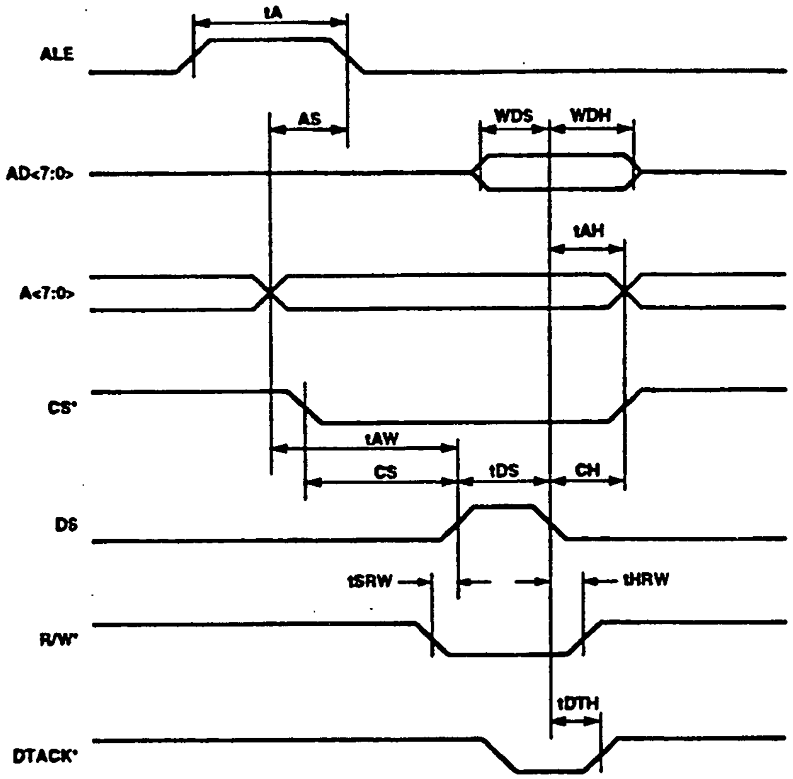
5.3.2.6 Register Read Operation in Motorola Nonmultiplexed Mode



5.3.2.7 Register Write Operation in Motorola Multiplexed Mode



5.3.2.8 Register Write Operation in Motorola Nonmultiplexed Mode

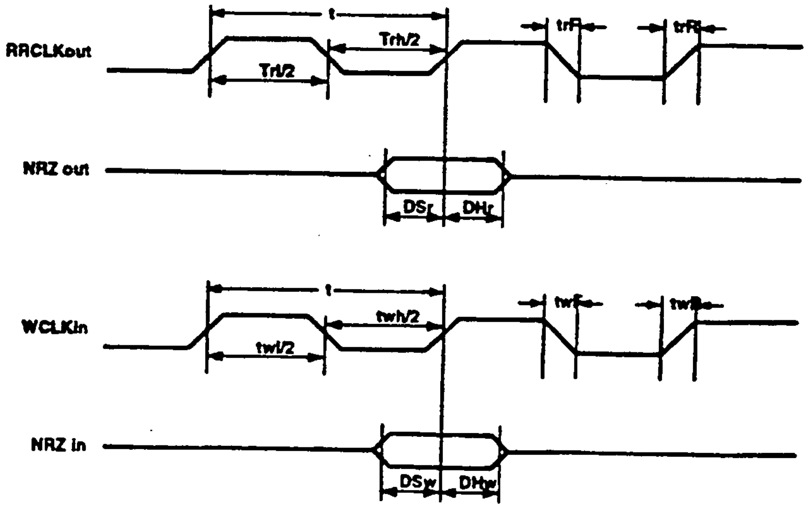


5.3.3 Disk Read/Write Timing Parameters

One(Two) Bit NRZ Interface

Symbol	Parameter	5v Min	5v Max	3.3v Min	3.3v max	Units
Tr	RRCLK period	25(50)		31(62)		ns
Trl/2(L)	RRCLK low time at 0.5V	10(23)		13(28)		ns
Trh/2(H)	RRCLK high time at 2.40V	10(23)		13(28)		ns
Tw	WCLK period	25 (50)		31(62)		ns
Twl/2	WCLK low time at 0.8V	8(20)		11(25)		
Twh/2	WCLK high time at 2.0V	8(20)		11(25)		
Trr,Trf	RRCLK rise and fall time		3		3	ns
Twr,Twf	WCLK rise and fall time		4		4	
Dsr	NRZ out valid to RRCLK out_	5			5	ns
Dhr	RRCLK out_ to NRZ out invalid	5			5	ns
Dsw	NRZ in valid to WCLK in	5			5	ns
Dhw	WCLK in to NRZ in invalid	5			5	ns

Disk Write Timing (One/Two Bit NRZ Interface)



Read

Write

6. SPECIFICATIONS

6.1 Gain Control

Acquisition Time	(48 to 372) * $1/f_S$	Typical: 48-84
Tracking Bandwidth	(0 to 7.5e-3) * f_S	

where f_S is the sampling rate.

6.2 Timing Recovery

Acquisition Time	(48 to 372) * $1/f_S$	Typical: 84-108
Tracking Bandwidth	(1e-3 to 2.5e-2) * f_S	

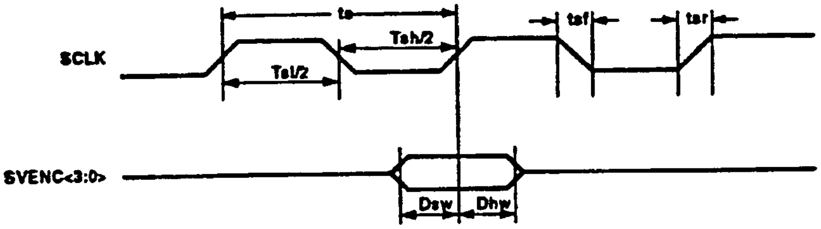
where f_S is the sampling rate.

7. INITIALIZATION CONDITIONS

<TBD>

5.3.4 Servo Bus Parameters

Symbol	Parameter	5v Min	5v Max	3.3v Min	3.3v max	Units
Ts	SCLK period					ns
Tsl/2(L)	SCLK low time at 0.5V					ns
Tsh/2(H)	SCLK high time at 2.40V					ns
Tsr,Tsf	SCLK rise and fall time					ns
Dsw	SVENC<3:0> valid to SCLK in					ns
Dhw	SCLK in to SVENC<3:0> in invalid					ns



0100:	1/64	1100:	Reserved
0101:	3/128	1101:	Reserved
0110:	1/32	1110:	Reserved
0111:	3/64	1111:	Reserved

8.4 Acquisition Timing Set Point/Coefficient 1

Bits 7-5: Acquisition Timing Recovery Coefficient 1. These bits determine the value of coefficient 1 (c_{1da}) used by the Timing Recovery compensation loop during data acquisition.

000:	Reserved	100:	1/8
001:	Reserved	101:	3/16
010:	1/16	110:	1/4
011:	3/32	111:	3/8

Bits 4-0: Acquisition Timing Set Point. These bits specify the value (t_a) used in computing phase errors during data acquisition. Range is -1 to +15/16.

8.5 Tracking Timing Set Point/Coefficient 1/Sample Phase Mode

Bits 7-5: Tracking Timing Recovery Coefficient 1. These bits determine the value of coefficient 1 (c_{1dt}) used by the Timing Recovery compensation loop during data tracking.

000:	1/32	100:	1/8
001:	3/64	101:	3/16
010:	1/16	110:	1/4
011:3	/32	111:	3/8

Bit 4: Sample Phase Mode

0: Generate samples for a center-sampled data pulse, i.e. a < b _ c < 1; timing is recovered using adjacent-but-one samples.

1: Generate samples for a side-sampled data pulse i.e. a _ c < b _ 1; timing is recovered using adjacent samples.

Bits 3-0: Tracking Timing Set Point. These bits specify the value (t_{dt}) used in computing is

8.6 Timing Recovery Coefficient 2

Bits 7-4: Acquisition Timing Recovery Coefficient 2. These bits determine the value of coefficient 2 (c_{2da}) used by the Timing Recovery compensation loop during data acquisition.

0000:	1/512	1000:	1/32

8. REGISTER DESCRIPTIONS

8.1 Acquisition Gain Set Point

Bits 7-6: Transition Threshold Bits 1-0. These bits specify the two least-significant bits of the Transition Threshold (V) used to qualify transitions during data tracking. Range is 0 to +3/16.

Bits 5-0: Acquisition Gain Set Point. These bits specify the value (g_a) used in computing gain errors during data acquisition. Range is 0 to +63/16.

8.2 Tracking Gain Set Point

Bits 7-6: Transition Threshold Bits 3-2. These bits specify the two most-significant bits of the Transition Threshold (V) used to qualify transitions during data tracking. Range is 0 to +3/4.

Bits 5-0: Tracking Gain Set Point. These bits specify the value (g_t) used in computing gain errors during data tracking. Range is 0 to +63/16.

8.3 Gain Control Coefficient

Bits 7-4: Acquisition Gain Control Coefficient. These bits determine the value of the coefficient (c_{gda}) used by the Gain Control compensation loop during data acquisition. Those values marked with * are likely to be useful only in side sampled mode. Those values marked with ** should not be used in side sampled mode.

0000:	Reserved	1000:	1/16
0001:	Reserved	1001:	3/32
0010:	Reserved	1010:	1/8
0011:	Reserved	1011:	3/16
0100:	Reserved	1100:	1/4
0101:	Reserved	1101:	3/8**
0110:	1/32*	1110:	1/2**
0111:	3/64*	1111:	Reserved

Bits 3-0: Tracking Gain Control Coefficient. These bits determine the value of the coefficient (c_{gdt}) used by the Gain Control compensation loop during data tracking. Those values marked with * are likely to be useful only in side sampled mode. Those values marked with ** should not be used in side sampled mode.

0000:	0	1000:	1/16
0001:	0	1001:	3/32**
0010:	1/128*	1010:	1/8**
0011:	3/256*	1011:	Reserved

0001:	3/1024	1001:	3/64
0010:	1/256	1010:	1/16
0011:	3/512	1011:	3/32
0100:	1/128	1100:	Reserved
0101:	3/256	1101:	Reserved
0110:	1/64	1110:	Reserved
0111:	3/128	1111:	Reserved

Bits 3-0: Tracking Timing Recovery Coefficient 2. These bits determine the value of coefficient 2 (c_{2dt}) used by the Timing Recovery compensation loop during data tracking.

0000:	1/4096	1000:	1/256
0001:	3/8192	1001:	3/512
0010:	1/2048	1010:	1/128
0011:	3/4096	1011:	3/256
0100:	1/1024	1100:	Reserved
0101:	3/2048	1101:	Reserved
0110:	1/512	1110:	Reserved
0111:	3/1024	1111:	Reserved

8.7 Timing Accumulator Initial Value

Bits 7-0: The value set in this register will be reloaded into the accumulator of the timing loop filter at the statr of each data acquisition.

8.8 Timing Accumulator read register

Bits 7-0: The most significant bits of the timing loop accumulator are loaded into this register at the end of acquisition. They can be loaded at other times by command

8.9 Gain Accumulator Initial value

Bits 7-0: Initial value of gain accumulator MSB's to be used at start of every data acquisition

8.10 Gain Accumulator read only

Bits 7-0: The most significant bits of the data gain loop accumulator are loaded into this register at the end of aquisition. They can be laoded at other times by command)

8.11 Offset Loop coefficient

Bit 7-4: Initial value of coefficient for offset filter for data acquisition mode.

Bit 3-0: Initial value of coefficient for offset filter for data tracking mode

8.12 Offset Accumulator Initial value

Bit 5-0:	Initial value of offset filter accumulator at start of data acquisition

8.13 Offset Accumulator - read only

Bit 5-0:	The most significant bits of the data and servo offset lloop accumulator are loaded into this register at the end of aquisition. They can be laoded at other times by command)

8.14 Digital Filter Coefficients: Taps 1 and 3

Bit 7-4:	Data Coefficient for tap # 1. May be varied from -6/16 to +6/16
Bit 3-0:	Data Coefficient for tap #3 . May be varied from -6/16 to +6/16

8.15 Digital Filter Coefficients: Tap #2 and gain

Bits 7-6:	Data Tap # 2 coefficient . May be set to 1/2, 3/4, 1, 5/4.
Bits 5-4:.	Data Filter gain. May be set to 1/2, 1,2,1
Bits 3-2:	Servo Tap # 2 coefficient . May be set to 1/2,3/4,1,5/4
Bits 1-0:	Servo Filter gain . May be set for 1.2,1,2,1
Reserved	111: Reserved

8.16 Spectral Smoothing Filter/Path Memory Control 1

Bits 7-6:	Header Detector Path Memory Length. These bits specify the length of the path memory in the sequence detector over headers. Headers are distinguished from data by the assertion of the Sequencer Output signal. 00: Six channel bits 01: Twelve channel bits 10: Eighteen channel bits 11: Twenty-four channel bits
Bit 5:	acquisition transition detector option 3a,3b for data
Bit 4:	sign of coefficients 1and 2
bit 3-2:	coefficient 2. May be set to 0,1/16,1/8

bits 1-0:	coefficient 1. May be set to 0,1/16,1/8,1/8

8.17 Spectral Smoothing filter/Path Memory Control 2

Bits 7-6:	Data Detector Path Memory Length. These bits specify the length of the path memory in the sequence detector over data. Headers are distinguished from data by the assertion of the Sequencer Output signal. 00: Six channel bits 01: Twelve channel bits 10: Eighteen channel bits 11: Twenty-four channel bits
Bit 5:	acquisition transition detector option 3a,3b for servo
Bit 4:	sign of coefficients 3 and 4
bit 3-2:	coefficient 4. May be set to 0,1/16,1/8
bits 1-0:	coefficient 3. May be set to 0,1/16,1/8,1/8

8.18 Spectral Smoothing filter Delay 1

Bits 4-0:	Spectral smoothing Filter Delay 1 (PD1). Range is 2 to 23.

8.19 Spectral Smoothing filter Delay 2

Bits 4-0:	Spectral smoothing Filter Delay 2 (PD2). Range is 2 to 23.

8.20 Detector Control 1

Bits 7-4:	Detector Level 3. These bits specify the last sample value (c) for a nominal pulse. Range is 0 to +15/16.
Bits 3-0:	Detector Level 1. These bits specify the first sample value (a) for a nominal pulse. Range is 0 to +15/16.

8.21 Detector Control 2

Bit 7:	Reserved
Bits 6-5:	MUX5 Control. These bits specify the source of the signal fed to the Sequence detector and Channel Quality circuit. 0x: Output of MUX3 10: Output of the SS Filter configured to handle only postcursor ISI

	11: Output of the SS Filter configured to handle both precursor and postcursor ISI
Bits 4-0:	Detector Level 2. These bits specify the second sample value (b) for a nominal pulse. Range is 0 to +31/16.

8.22 Acquisition Length/Multiplexer Control

Bit 7:	MUX4 Control. This bit controls the output of MUX4. 0: Output of Sequence Detector. 1: Output of Transition Detector.
Bit 6:	MUX3 Control. This bit controls the output of MUX3 during tracking; during acquisition, MUX3 selects raw input samples. 0: Raw input samples. 1: Delayed or Filtered input samples, as controlled by the MUX1 Control and MUX2 Control bits.
Bit 5:	MUX2 Control. This bit controls the output of MUX2. Note that the output of MUX2 is not used during acquisition. 0: Filtered input samples. 1: Delayed input samples.
Bit 4:	MUX1 Control. This bit controls the output of MUX1. Note that the output of MUX1 is not used during acquisition. 0: Filtered input samples. 1: Delayed input samples.
Bits 3-0:	Acquisition Length for data. This value controls the duration, in NRZ byte times, of acquisition for Gain Control and Timing Recovery. The minimum value which should be used is four. Note that the total length of the preamble is controlled by the Disk Controller and should be longer than the Acquisition Length by at least one byte.

8.23 Mux control Acquisition mode

Bit 7:	MUX3 Control for data acquisition.
Bit 6:	MUX2 Control for data acquisition
Bit 5:	MUX1 Control for data acquisition.
Bit 4:	MUX3 Control for servo acquisition
Bit 3:	MUX2 control for servo acquisition

Bit 2:	MUX 1 control for servo acquisition
Bits 1-0:	Reserved for future use

8.24 Control/Status Register

Bit 7:	Chip Reset. Assertion of this bit places the SH3300 in a reset state. This bit must be written with zero before the SH3300 can be programmed or used.
Bit 6-5:	Power Management Modes: Read Write, track following, idle
Bit 4:	Reserved - should be set to zero on write - value undefined on read
Bits 3:	Reserved - should be set to zero on write - value undefined on read
Bit 2:	Gain loop saturation flag. Set to one if saturation occurs, reset when this register is written.
Bit 1:	Timing loop saturation flag. Set to one if saturation occurs, reset when this register is written.
Bit 1:	Acquisition Timing Recovery Register Saturation. This bit is set if the Acquisition Timin g Recovery register saturates. It is reset when this register is written.
Bit 0:	Tracking Timing Recovery Register Saturation. This bit is set if the Tracking Timing Recovery register saturates. It is reset when this register is written.

8.25 Endec/NRZ/Gate Control

Bit 7:	Read/Write Gate Polarity Select 0: RG/RG* and WG/WG* are positive-true. 1: RG/RG* and WG/WG* are negative-true.
Bit 6:	Erasure Pointer Enable 0: Normal operation; data bits are output on NRZ bus. 1: Erasure pointers are output on NRZ bus.
Bits 5-4:	Data Randomizer Control x0: Data Randomizer feedback is enabled. x1: Data Randomizer feedback is disabled; the initial contents of the Data Randomizer consitute a two NRZ-byte repeated pattern. This bit should be set only when the Channel Quality Mode is set. 0x:Initialize Data Randomizer to zero. 1x:Initialize Data Randomizer to Data Randomizer Seed.
Bit 3-2:	Reserved for future use
Bits 1-0:	NRZ Mode Select 00: One-bit NRZ Mode is used.

01: reserved
10: Two-bit NRZ Mode is used.

11: reserved.

8.26 Synchronization Mark Control

Bits 7-6:	Reserved.
Bits 5-4:	Synchronization Mark Length. These bits specify the length of the error-tolerant synchronization mark. 01: One NRZ byte (twelve channel bits). 10: Two NRZ bytes (twenty-four channel bits). 11: Three NRZ bytes (thirty-six channel bits). 00: Four NRZ bytes (forty-eight channel bits).
Bits 3-0:	Synchronization Mark Threshold. These bits specify the number of four-channel-bit groups in the Synchronization Mark Pattern which must be detected without error in order to detect the Synchronization Mark. Range is 0 to three times the Synchronization Mark Length.

8.27 Synchronization Mark Recovery Count

Bits 7-0:	Synchronization Mark Recovery Count. These bits specify the number of channel bits to wait before assuming that the Synchronization Mark has been detected and beginning to decode data.

8.28 Synchronization Mark Pattern 1

Bits 7-0:	Synchronization Mark Pattern 1.These bits specify the first eight channel bits of the Synchronization Mark Pattern.

8.29 Synchronization Mark Pattern 2

Bits 7-0:	Synchronization Mark Pattern 2. These bits specify the second eight channel bits of the Synchronization Mark Pattern.

8.30 Synchronization Mark Pattern 3

Bits 7-0:	Synchronization Mark Pattern 3. These bits specify the third eight channel bits of the Synchronization Mark Pattern.

8.31 Synchronization Mark Pattern 4

Bits 7-0:	Synchronization Mark Pattern 4. These bits specify the fourth eight channel bits of the Synchronization Mark Pattern.

8.32 Synchronization Mark Pattern 5

Bits 7-0:	Synchronization Mark Pattern 5. These bits specify the fifth eight channel bits of the Synchronization Mark Pattern.

8.33 Synchronization Mark Pattern 6

Bits 7-0:	Synchronization Mark Pattern 6. These bits specify the sixth eight channel bits of the Synchronization Mark Pattern.

8.34 Data Randomizer Seed LSB

Bits 7-0:	Data Randomizer Seed LSB. These are the eight least-significant bits of the Data Randomizer seed.

8.35 Data Randomizer Seed MSB

Bits 7-0:	Data Randomizer Seed MSB. These are the eight most-significant bits of the Data Randomizer seed.

8.36 Channel Quality Control 1

Bit 7:	Channel Quality Mode. When this bit is written with '1', the Channel Quality Count and Output registers are cleared. 0: Normal Operation.

1: Channel Quality Mode.

Bit 6: Reserved.

Bit 5: Channel Quality Sign Enable
0: Channel Quality Sign State bits are ignored.
1: Channel Quality Sign State bits are used to qualify Channel Quality accumulation. When this bit is set, one and only one of the Channel Quality Error Input Control bits should be set. The purpose of this option is to allow independent measurement of the properties of positive and negative transitions.

Bits 4-0: Channel Quality Sign State. These bits define the sign state of the MLSD used to qualify Channel Quality accumulation. The mapping between each of the positive and negative combinations described by the Channel Quality Error Input Control bits and the Channel Quality Sign State is <TBD>.

8.37 Channel Quality Control 2

Bits 7-5: Channel Quality Input Source. This bit controls whether squared sample errors or squared gain/timing errors, or unsquared samples are accumulated by the Channel Quality circuit.
00x: Squared EVEN sample errors plus squared ODD sample errors are accumulated.
010: Squared Gain errors are accumulated.
011: Squared Timing errors are accumulated.
1x0: Unsquared EVEN samples are accumulated.
1x1: Unsquared ODD samples are accumulated.

Bits 4-0: Channel Quality Input Control 12-8. These bits independently enable Channel Quality accumulation when the given combination of nominal sample values (and/or the corresponding negative combination, as controlled by the Channel Quality Sign Enable bit) are expected.

	EVEN	ODD
xxxx1:	c	-a
xxx1x:	b	1-a
xx1xx:	b-c	1-a
x1xxx:	b	1
1xxxx:	b-c	1

8.38 Channel Quality Control 3

Bits 7-0: Channel Quality Error Input Control 7-0. These bits independently enable samples corresponding to the given combination of nominal sample values (and/or the corresponding negative combination, as controlled by the Channel Quality Sign Enable bit) to be accumulated in the Channel Quality Output.

EVEN ODD

xxxxxxx1:	0	0
xxxxxx1x:	1	c
xxxxx1xx:	c	0
xxxx1xxx:	a	b
xxx1xxxx:	c-a	-b
xx1xxxxx:	1-a	c-b
x1xxxxxx:	0	a
1xxxxxxx:	1	c-a

8.39 Channel Quality Count LSB

Bits 7-0:	Channel Quality Count LSB. This value is the eight least-significant bits of th number of clocks over which the Channel Quality Output was accumulated.

8.40 Channel Quality Count MSB

Bits ?-0:	Channel Quality Count MSB. This value is the most-significant bits of the numbe of clocks over which the Channel Quality Output was accumulated.

8.41 Channel Quality Output LSB

Bits 7-0:	Channel Quality Output LSB. This value is the eight least-significant bits of the Channel Quality Output word.

8.42 Channel Quality Output MID

Bits 7-0:	Channel Quality Output MID. This value is the second eight least-significant bits of the Channel Quality Output word.

8.43 Channel Quality Output MSB

Bits ?-0:	Channel Quality Output MSB. This value is the most-significant bits of the Channel Quality Output word.

8.44 Gain Control Loop Register

Bits 7-0:	Gain Control Loop Register. The value read and written here is the eight most-sig-

nificant bits of the Gain Control Loop register. When this register is written, the remaining bits of the Gain Control Loop register are cleared.

8.45 Acquisition Timing Recovery Loop Register

Bits 7-0: Acquisition Timing Recovery Loop Register. The value read and written here is the eight most-significant bits of the Acquisition Timing Loop Recovery Loop register. When this register is written, the remaining bits of the Acquisition Timing Recovery Loop register are cleared.

8.46 Tracking Timing Recovery Loop Register

Bits 7-0: Tracking Timing Recovery Loop Register. The value read and written here is the eight most-significant bits of the Tracking Timing Recovery Loop register. All bits of this register are automatically cleared at the end of acquisition. When this register is written, the remaining bits of the Tracking Timing Recovery Loop register are cleared. All bits of the Tracking Timing Recovery Loop register are automatically cleared at the end of acquisition.

8.47 Synthesizer Numerator

Bits 7-0: Synthesizer Numerator. This is the m value used in synthesizing the reference clock frequency as m/n times the REFCLK frequency.

8.48 Synthesizer Denominator

Bits 5-0: Synthesizer Denominator.. This is the n value used in synthesizing the reference clock frequency as m/n times the REFCLK frequency.

8.49 Frequency VCO Control

Bits 7-4: VCO Center Frequency. This value controls the center frequency of the VCO for data.

Bits 3-0: VCO center Frequency. This value controls the center frequency of the VCO for servo.

8.50 Analog Filter Boost Control

Bits 7-4:	Analog filter boost control in steps of 0.75dB up to 12dB.for use with data
Bits 3-0:	Analog Filter Boost control in steps of 0.75dB up to 12dB for use with servo

8.51 Analog Filter Cutoff Control (data)

Bits 7-0:	Analog filter cutoff control for data

8.52 Analog Filter Cutoff control (servo)

Bits 7-4:	Analog filter cutoff control for servo

8.53 VGA Control

Bits 7-4:	VGA Coarse Gain. This value controls the coarse gain of the VGA.
Bits 3-0:	VGA DC Offset. This value is fed to a DAC which generates an offset bias current for the VGA. Nominal value is <TBD>.

8.54 Comparator Control

Bits 7-4:	Reserved - should be set to zero on write - value undefined on read
Bits 3-0:	Hysteresis Comparator Data. These bits control the hysteresis comparator during data operations.

8.55 Precompensation Control

Bits 7-0:	Write Precomp Time Unit. These bits determine the time delay unit used for write precompensation.

NOTE: The write precompensation strategies required for a minimum run-length constraint of zero are more complex than those required for a minimum run-length constraint of one.

8.56 Power Control

Bit 7:	Test Mode Enable 0: Normal Operation 1: Test Mode
Bit 6:	Reserved
Bit 5:	Preamp Idle Mode 0: The preamp is enabled for normal operation. 1: The preamp is disabled to allow proper VGA DC Offset to be determined.
Bit 4:	Write PLL (Synthesizer) Power On.
Bit 3:	VGA/Filter Power On.
Bit 2:	VGA/Filter Power On if Read Gate is asserted or servo burst mode is active.
Bit 1:	ADC/Read VCO Power On.
Bit 0:	ADC/Read VCO Power On if Read Gate is asserted.

8.57 Mode Control

Bits 7-6:	VGA Disable Time. These bits specify how long the VGA inputs remain tied together after the trailing edge of Write Gate. 00: 0 reference clock periods. 01: 8 reference clock periods. 11: 16 reference clock periods. 11: 32 reference clock periods.
Bit 5:	Write Precompensation Enable 0: Write precompensation is disabled. 1: Write precompensation is enabled.
Bit 4:	Write Precompensation Linearity 0: Linear write precompensation is selected. 1: Non-linear write precompensation is selected.
Bit 3:	Write Precompensation Constraint 0: Write precompensation for run-length constraint of zero is selected. 1: Write precompensation for run-length constraint of one is selected.
Bit 2:	Sample Phase Mode 0: Controlled-phase start is enabled for center-sampled pulses, i.e. a < b _ c < 1. 1: Controlled-phase start is enabled for side-sampled pulses, i.e. a _ c < 1, b _ 1.
Bit 1:	Servo Output Select

5. ELECTRICAL SPECIFICATIONS

5.1 Absolute Maximum Ratings (For 5v and 3.3v devices unless stated otherwise)

Parameter	Minimum	Typical	Maximum	Units
Ambient Temperature Under Bias	0		70	C
Storage Temperature	-65		150	_C
Voltage On Any Pin	-0.5		VCC+0.5	V
Power Dissipation SH3300		400		mW
Power Dissipation SH3303		170		mW
Power Supply Voltage (5v device)			7.0	V
Power Supply Voltage (3.3v device)				

NOTE: Stresses above those listed may cause permanent damage to the device. This is a stress rating only, and functional operation of the device at these conditions, or any conditions outside those indicated in the operational sections of this specification, is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

5.2 DC Characteristics (digital)

	5v	5v	3v	3v	Units
Parameter	Min	Max	Min	Max	Units
Supply Voltage	4.5	5.5	3.0	3.63	V
Input Low Voltage	-0.5	0.8	-0.3	+0.8	V
Input High Voltage	2.0	VCC+0.5	2.0	Vcc+0.3	V
Output Low Voltage @ IOL = 2.0 mA		0.4		0.45	V
Output High Voltage @ IOH = -400 µA	3.5		Vcc-0.2		V
Supply Current		<TBD>		<TBD>	mA
Input Leakage Current	-10	+10	-10	+10	µA
Output Leakage Current		+/-15		+/-15	µA
Input Capacitance		10		10	pF
Output Capacitance		10		10	pF

NOTE: All unused inputs must be tied to GND or VCC

9. GLOSSARY

Aliasing - The phenomenon whereby energy at frequencies above one-half the sampling rate appears as energy at frequencies below one-half the sampling rate in a sampled signal.

Antialiasing Filter - A filter which attenuates signal energy above a frequency of one-half the sampling rate.

Autocorrelation Function - The integral _(_) of the product of a function x(t) and x(t+_); e.g _(_) = E(x(t)_x(t+_)). For a binary sequence, this can be treated as the number of bits matching between two copies of a sequence as a function of the offset between the sequences.

Calibration - The process whereby the analog and digital parameters of the channel and the detector are matched for best performance.

Channel Constraint - A limitation placed on the sequence of channel bits. For example, a RLL (d,k) channel constraint enforces a minimum (d) and maximum (k) number of consecutive '0's between '1's.

Channel Quality - A measure of how well matched are a channel and detector.

Channel Code - A set of rules governing the encoding of data bits to channel bits. Channel codes typically enforce a minimum run-length constraint (to limit non-linear effects) and/or a maximum run-length constraint (to aid in timing recovery).

Code Rate - The ratio of the number of bits in the data-bit stream to the number of bits in the channel-bit stream. For a RLL (1,7) code, the code rate is 2/3.

Coding Gain - The product of the code rate and the minimum distance of the encoded sequences of a code.

Convolution - The convolution y(k) of sequences u(k) and h(k) is given by
y(k) = _ u(m) * h(k - m)
where the summation is over all m. When u(k) is the input to a filter whose impulse response is h(k), the filter output is the convolution y(k).

Correlation Function - The expected value of the product of two random variables, which may be scalars or vectors.

Dicode - Another name for Partial Response Class I (PR1).

Distance - A measure of the "difference" between two vectors.

Duobinary - The partial response channel corresponding to the polynomial (1 + D).

Error Propagation - The process wherein a channel decoder emits more than one erroneous data bit due to a single channel bit in error or more generally a burst of erroneous data bits which is longer (after accounting for code rate) than the causative channel-bit error burst.

Error Tolerant Synchronization Mark - A synchronization mark which can be detected in the presence of error with an acceptable false detection probability.

Euclidean Distance - The square root of the sum of the squares of the differences in each dimension between two n-dimensional vectors.

Excess Bandwidth - Signal energy passed by a filter at frequencies above one-half the sampling rate.

Extended Partial Response Class IV (EPR4) - The partial response channel corresponding to the polynomial $(1 - D)*(1 + D)^2$.

Eye Diagram - A graphical representation of the effects of intersymbol interference. It illustrates the typical or maximum excursion of a signal from its ideal sample levels at its ideal sampling instants. It is important to note that MLSD is not dependent on the eye being open in the same way that sample-by-sample detection is.

Least-Mean-Square Algorithm - A procedure used in adapting the coefficients of a digital filter in which the objective is to minimize the mean-square error of the filter output.

Maximum A Posteriori Probability Sequence Detection (MAP) - A decoding method in which the channel bit sequence emitted from the detector is that which, given the channel sample sequence, is the most probable written sequence. MAP maximizes the probability density function of the detector output given the detector input.

Maximum Likelihood Sequence Detection (MLSD) - A decoding method in which the channel bit sequence emitted from the detector is that with the highest probability of having produced the channel sample sequence input to the detector. MLSD maximizes the probability density function of the detector input given any possible channel input. MLSD is equivalent to MAP if the probabilities of all possible written sequences are equal or if they are assumed to be equal when no information is available concerning the probabilities of written sequences.

Mean-Square Error - A measure of error in which the difference between each sample and its ideal sample level is squared and summed. The sum divided by the number of samples is the mean-square error of the sequence.

Metric - A function of two vectors satisfying certain properties that allow it to be interpreted as a measure of distance.

Minimum Distance - The smallest distance between any two vectors in a set of vectors.

Modified Duobinary - Another name for Partial Response Class IV (PR4).

Noise Types:

Coherent Noise - Noise which is correlated with the signal e.g. transition noise.

Colored Noise - Noise which is correlated between one sample instant and another.

Gaussian Noise - Noise whose amplitude has a normal probability distribution.

Misequalization Noise - Amplitude errors which remain at the output of an equalizer due to its imperfect pulse-shaping. This is something of a misnomer, as this "noise" source is predictable for a given channel transition response, equalizer, and transition sequence.

Offtrack Noise - When tracking is not perfect and the head deviates from the center of a track, the amplitude of the signal from the target track is reduced and the head picks up signal energy due to transitions on the adjacent track.

Out-of-band Noise - Noise with energy at frequencies above one-half the sampling rate.

Overwrite Noise - When erasure of previously written transitions is not perfect, the alignment of magnetic domains is not complete and uniform and the head responds to the deviations.

Preamp Noise - Electronic noise arising in the first stage of the head preamplifier. Due to the low signal amplitude present at the input to the preamp and the correspondingly high gain required, it is possible for preamp noise to be the dominant noise source in a read channel.

Synchronous Noise - An interfering source with a frequency equal to the sample rate or a multiple thereof. The energy of such a source is aliased to DC and can be canceled by adjusting the VGA DC Offset of the Galaxy-A.

Transition Noise - Coherent noise caused by "zig-zag" transitions occurring on thin-film media.

White Noise - Noise which is not correlated between one sample instant and another.

Nonlinear Transition Shift - Movement in the effective location of a transition toward a very near preceding transition. This is caused by demagnetization during write.

Norm - A particular method of measuring the lengths of vectors; related to a metric.

NRZ - Non-Return to Zero refers to a bit sequence to which no run-length constraint applies.

NRZI - Modified Non-Return to Zero refers to a bit sequence to which no run-length constraint applies but in which each '1' or '0' represent the presence or absence, respectively, of a (magnetic) transition. NRZI sequences can be produced from NRZ sequences via a precoding process equivalent to dividing by (1 D).

Partial Response Channel - A channel in which controlled intersymbol interference is allowed. In such a channel, symbols (e.g. transitions in a magnetic recording channel) are so closely spaced that the channel begins to respond to the next symbol when only part of its response to the current symbol has occurred.

Partial Response Class I (PR1) - The partial response channel corresponding to the polynomial (1 - D).

Partial Response Class IV (PR4) - The partial response channel corresponding to the polynomial (1 - D)*(1 + D).

Path Metric - A cumulative measure of error maintained for a path through a decoder's state machine based on a particular norm.

Preamble - A special repetitive pattern which is used to allow gain to be adjusted to an appropriate level and frequency and phase lock to be acquired. The SH3300 can use a preamble pattern of '1010...' or '100100...' or '100010000...'.

Precoding - The process of encoding a sequence to account for some known property of a channel e.g. the differentiating nature of a magnetic recording channel.

Raised Cosine - A form of equalization which is a more-realizable modification of the "brick wall" filter which would produce a sinc pulse. It is characterized by a transfer function H(f) defined as:

H(f) = 1 for 0 < f < (1 - _)*f_s/2
= 1/2*[1 + cos(_/[_*f_s]*[f - (1 - _)*f_s/2])] for (1 - _)*f_s/2 _ f _ (1 + _)*f_s/2
= 0 for f > (1 + _)*f_s/2

where f_s is the sample rate and _ is a parameter in the range zero to one.

Run-Length Limited Code - A code in which the minimum and maximum numbers of consecutive '0's and/ or '1's is constrained by a systematic rule. The notation RLL (d,k) means that no less than d and no more than k consecutive '0's are allowed in the encoded bit stream, where '0' and '1' usually represent the absence or presence, respectively, of a (magnetic) transition.

Run-Length Violation - An case in which a defect or noise event has caused the occurrence of a sequence of '0's and '1's which violates the run-length constraints of the code.

Split Field - A recording scheme in which the data/redundancy area of a sector is divided into two or more areas, typically by one or more areas used for tracking control in an embedded-servo disk drive.

State Machine - A circuit in which the present state of a channel or system is represented by a number of bits whose values are controlled by the present values of the bits and the values of external signal levels or the occurrence of external signal transitions.

Synchronization Mark - A pattern, preferably unique and different from any valid encoded channel-bit sequence, used to signify the start of a header or data area. A good synchronization mark has low cross-correlation with the preamble and small autocorrelation function values at non-zero offsets, implying a low false detection probability.

11. PACKAGE INFORMATION

SH3300 will be available in a 64 pin SQFP package. Package drawing TBD

BACKGROUND OF THE INVENTION

In a computer data storage device, data is typically stored on magnetic media by reversing the magnetic flux at each "one" bit location, and maintaining the same flux direction at each "zero" bit location, called NRZI recording. When these flux reversals pass a read head they cause a voltage change in the read head. The voltage from the various flux reversals along a data track appears as a varying signal level on the output of the read head, with the voltage caused by each flux reversal appearing as a positive or negative voltage pulse. These pulses must be detected by the electronics connected to the read head, typically by detecting the peak of each pulse to determine the pulse location.

Similarly, the voltage from a data transmission line appears as a varying signal level on the input to a data communication device, such as a modem, with the voltage of each bit change often appearing as a positive or negative voltage pulse.

In order to separate the voltage pulses from noise in the signal, the voltage may be compared to a predetermined threshold, and only peaks whose magnitude exceeds the threshold are considered to be possible pulses. If the peaks are too small with respect to the threshold, the signal gain must be adjusted higher, and if the peaks are too high the gain must be adjusted lower. To properly determine the gain setting, pulses must be detected so that the gain adjustment can maintain the proper level for a pulse peak.

Because the rotational speed of a disk may vary, a phase locked loop is used to synchronize a detector to the times of the pulses. The peaks of the pulses caused by bit transitions must be detected in order to correct for any timing deviations.

Several methods have been used to detect the pulses and thus the transitions within the data. In the prior art, peak detection circuitry has ordinarily been analog in nature, while digital detectors have ordinarily been sequence detectors such as those implementing the Viterbi method, disclosed in Application Serial Number 07/852,015, filed March 16, 1992, of Richard T. Behrens, Kent D. Anderson and Neal Glover, entitled "Method and Apparatus for Reduced-Complexity Viterbi-Type Sequence Detectors", which is incorporated herein by reference for all that is disclosed and taught therein. However, most digital pulse detectors analyze the context of the pulses, and do not detect a pulse until several samples have been taken beyond the peak of the pulse. This latency in detecting pulses is a disadvantage when the location of a pulse needs to be known in order to make timing and/or gain corrections.

It is thus apparent that there is a need in the art for an improved apparatus which digitally detects pulses at the earliest possible time after the peak of the pulse occurs. The present invention meets these and other needs.

SUMMARY OF THE INVENTION

It is an aspect of the present invention to detect signal pulses caused by bit transitions within data being read from a data storage device.

It is another aspect of the invention to detect signal pulses within signals received by a data communications device.

It is another aspect of the invention to detect the pulses using samples of the signal wherein one of the samples occurs at a peak of the pulse caused by a bit transition.

Another aspect of the invention is to detect such pulses using samples of the signal wherein samples occur on each side of a peak of the pulse and wherein no sample occurs at the peak.

A further aspect is to detect a pulse within one half to one and one half sample periods from the peak of the pulse.

A still further aspect is to use signal level moving averages of two samples for detection.

The above and other aspects of the invention are accomplished in a digital pulse detector that uses four samples of a read signal to detect a data transition as soon as one sample beyond the time of the peak of the signal level at the bit transition. The pulse detector detects peaks by sampling at the center of the peak or by sampling at each side of the peak. The pulse detector detects pulses while tracking, where the pattern of pulses is not known in advance, or while acquiring timing lock on the signal in a preamble, where the data pattern is fixed. Thus the pulse detector provides four combinations of detection to detect in tracking or acquisition mode, using either center or side sampling in either mode. The detection rules are summarized in Table 2.

In an alternative embodiment, the detector uses a moving average of two adjacent samples, as well as individual sample signal levels, to perform the detection. The alternative embodiment uses the sampled signal levels directly to determine peaks in side sampled acquisition mode, and in center sampled tracking mode. In the other two of the combinations, the alternative embodiment detector uses two-sample moving averages to determine peaks. The preamble pattern for acquisition in the alternative embodiment is restricted to be a repeating sequence of alternating polarity pulses with a period of four sample intervals. This is more restrictive than the pattern handling capability of the first embodiment, however, it permits the alternative embodiment to perform earlier detection of the pulses in the side sampled acquisition case and it uses simpler rules in acquisition. The detection rules for the alternative embodiment are summarized in Table 1.

Both embodiments avoid using a signal threshold to qualify the pulses during acquisition. This aspect prevents the gain at the beginning of the acquisition from affecting acquisition, since the gain may be incorrect at the beginning of acquisition. During tracking, where the pulses may be more widely separated, a threshold is used to qualify pulses. This qualification prevents the detection of small noise-induced pulses as data pulses.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other aspects, features, and advantages of the invention will be better understood by reading the following more particular description of the invention, presented in conjunction with the following drawings, wherein:

Fig. 1 shows a block diagram of the invention and the environment of the invention;

Fig. 2 shows a block diagram of the circuitry that provides data samples to the invention;

Fig. 3 shows a block diagram of the digital circuitry of the read channel containing the invention;

Fig. 4 shows a signal waveform and illustrates center sampling of the pulses;

Fig. 5 shows a signal waveform and illustrates side sampling of the pulses;

Figs. 6, 7, and 8 show a high-level logic diagram of the pulse detector of the present invention for the equations of Table 1; and

Figs. 9, 10, and 11 show a high-level logic diagram of the pulse detector of the present invention for the equations of Table 2.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The following description is of the best presently contemplated mode of carrying out the present invention. This description is not to be taken in a limiting sense but is made merely for the purpose of describing the general principles of the invention. The scope of the invention should be determined by referencing the appended claims.

Fig. 1 shows a block diagram of the invention and a typical environment of the invention. Referring now to Fig. 1, a computer system 100 contains a processing element 102 which communicates to other elements of the computer system 100 over a system bus 104. A keyboard 106 and a display 108 allow a user of the computer system 100 to communicate with the computer system 100. A memory 110 contains programs and data which cause the computer system 100 to perform operations desired by the user.

A disk data storage system 112 is connected to the system bus 104 for storing data and programs within the computer system 100. A disk controller 114 within the disk device 112 communicates to the system bus 104 and controls the operations of a disk drive 118, possibly in conjunction with a local microprocessor (not shown) within the disk data storage system 112. The disk drive 118 performs the storage function, typically storing the data on magnetic media. A bus 116 connects the disk controller 114 to the disk drive 118, specifically connecting to a write channel 120 to write data onto the disk through write heads and amplifiers 128. When data is being read from the disk through the read head and amplifiers 128 the data comes back through the read channel 122 which contains the pulse detector of the present invention. The read and write heads may be physically the same heads. The data first passes through the analog section 126 of the read channel 122 and then through the digital section 124 of the read channel 122 before being sent on the bus 116 to the disk controller 114. After being processed by the disk controller 114, the data is then sent over the system bus 104 to the memory 110 where it is processed.

The disk controller 114 also connects to other circuits, not shown, within the disk drive 118, such as a circuit which moves the read/write heads over the surface of the data storage media.

Although not shown in Fig. 1, the pulse detector of the present invention can be used to detect pulses within data received from a transmission line, such as a telephone line or local area network, in a data communications receiver. It may also be used in any other device that must detect pulses within a signal.

Fig. 2 shows a block diagram of the analog circuitry 126 of the read channel 122. Referring now to Fig. 2, when a read head is passing over a track of the data storage medium, it picks up a signal which is amplified by a preamplifier, not shown. After this preamplification, the signal 201 is passed to a variable gain amplifier 202. The signal is further amplified by the variable gain amplifier 202 and passed through an analog equalizer circuit 204, which filters the signal as desired, for example, so as to remove unwanted high frequencies and shape the remaining spectrum, to an Analog to Digital converter 206. The A to D converter 206 converts the analog signal into a digital value, providing six bits of digital information in the preferred embodiment, and then the data is passed to a register/de-multiplexer 207. In the preferred embodiment, the digital section 124 of the read channel 122 processes two samples in parallel. To create these two samples, the register/de-multiplexer 207 stores every other sample taken by the A to D converter 206. After the second sample is taken, the data from the two samples is passed to the data bus 230. The bus 230 is clocked by a single half-frequency clock signal.

The timing necessary for converting the data, also called taking a sample, in the A to D converter 206 is supplied by a variable frequency oscillator 222 which is controlled by the output of a digital to analog converter (DAC) 220. The input to the DAC 220 comes from the digital section of the read channel 124 as timing feedback signal 234.

The gain of the variable gain amplifier 202 is controlled through a gain feedback signal 232 which originates in the digital portion 124 of the read channel 122. The gain feedback signal 232 is input to a summing junction 210 which has a coarse gain control value as its other input. The coarse gain control can be set by the disk controller 114, or a local microprocessor within the disk drive (not shown), to provide a nominal gain level which is then adjusted up or down by the gain feedback signal 232. After being summed with the coarse gain control value, the feedback signal is sent to a digital to analog converter 212 and then to a filter 214. Because of the nature of digital to analog converters, the output of the DAC 212 may contain glitches when it is changing values. Therefore, the filter 214 may be necessary to remove these glitches in the feedback signal. After being filtered, the signal is converted to an exponential value by the exponential converter block 216 and then connected to the variable gain amplifier 202. This conversion makes the small-signal gain control dynamics independent of the input signal amplitude.

Fig. 3 shows a block diagram of the digital section 124 of the read channel 122 (Fig. 1). Referring now to Fig. 3, the digital data signal 230 from Fig. 2 is input to a delay circuit 304 and a digital filter circuit 302. The digital data signal 230 is also input to a multiplexer 306 whose output is connected to a second multiplexer 310 with an output that feeds the pulse detector 312 of the present invention. The output of the pulse detector 312 is connected to a gain control circuit 330 which provides the gain feedback signal 232 that connects to Fig. 2. The output of the pulse detector 312 is also connected to a timing recovery circuit 328 whose output 234 connects to the digital to analog converter 220 of Fig. 2. The output of the pulse detector 312 of the present invention may also be connected to a sync mark detector 322 and an RLL decoder 320, as shown in Fig. 3, or a more sophisticated data detector (not shown) may be connected to the sync mark detector 322 and the RLL decoder 320. The output of the RLL decoder 320 and the sync mark detector 322 are connected to the disk controller 114 (Fig. 1) through the bus 116.

The pulse detector 312 of the present invention is designed to detect pulses using one of two types of sampling methods, selected by the user. The first detection method is called center sampling wherein one of the samples taken will arrive at or very near the center, or peak, of a pulse. The location of the sampling is controlled by the timing recovery block 328 of Fig. 3. In the other method of sampling, called side sampling, the timing recovery block 328 will adjust the timing of the VFO 222 (Fig. 2) such that two samples are taken wherein one of the two samples appears on one side of the peak of the pulse and the other of the two samples appears on the other side of the peak. The user of the system determines whether center sampling or side sampling is used by setting a bit in a control register through the interface 116 (Fig. 1).

For either center or side sampling, sampling can occur at two different times within a data record being read from the disk media. The first time sampling occurs is called acquisition, because it occurs when the gain control and timing control are acquiring the gain and timing relationships of the pulses. This occurs when the read head is passing a preamble portion of the data record which always has a known data pattern to facilitate acquisition of the timing and gain. When data is being transferred, a different mode is used for the pulse detector, called tracking, since data has an irregular and a priori unknown pattern of pulses.

Therefore, the pulse detector of the present invention is designed to detect pulses under four separate conditions. The first condition is acquisition mode using side sampling, the second is acquisition mode using center sampling, third is tracking mode using side sampling, and fourth is tracking mode using center sampling.

The pulse detector analyzes the current sample of the amplitude of the signal as well as the previous three samples of the signal amplitude. Using these four samples, Table 1 shows equations for one embodiment of the pulse detector, and Table 2 shows equations for another embodiment.

In Table 1, y_n is the current sample, y_{n-1} is the first previous sample, and y_{n-2} is the second previous sample. The third previous sample is used only through the moving averages, (1+D)/2 (described below). The equations of Table 1 will be further described below with respect to Figs. 5 and 6. Also, the equations of Table 1 show pulse detection of a pulse whose peak occurs at time y_{n-1} for center sampling, or between y_{n-1} and y_{n-2} for side sampling tracking mode, or between y_n and y_{n-1} for side sampling acquisition mode. The circuits of Figs. 6-8 processes two signal samples simultaneously. The equations for the second sample are the same as the equations of Table 1, with one time delay. Thus, for example, y_n in Table 1 would be replaced by y_{n-1}, y_{n-1} would be replaced by y_{n-2}, and y_{n-2} would be replaced by y_{n-3}.

Fig. 4 shows a signal waveform of an isolated pulse and illustrates center sampling of the pulse. Referring now to Fig. 4, a signal waveform 402 is shown having a positive level above a baseline 404. Four samples of this waveform have been taken with the sample identified by reference 412, y_n, being the most recent sample. Sample 410, y_{n-1}, is the sample just prior to the most recent sample, sample 408 is the next previous sample and sample 406 is the oldest of the four samples shown. A threshold value 414 is also shown and the signal value of sample 410 must be greater than the threshold 414 for the pulse to be detected in tracking mode. Negative pulses would appear as a mirror image of Fig. 4.

Fig. 5 shows a signal waveform of an isolated pulse and illustrates side sampling of the pulse. Referring now to Fig. 5, a signal waveform 502 is shown as a positive level above a baseline 503. Four samples are shown, with the most recent sample being sample 510. Sample 508 is the sample previous to the most recent, 506 is the next previous sample and sample 504 is the oldest of the four samples shown. These samples are also identified by the reference y_n through y_{n-3}. Also, a threshold 512 is illustrated, and sample $(y_{n-1} + y_{n-2}) / 2$ must be greater than the threshold 512 to be considered a pulse during tracking.

Figs. 6, 7, and 8 show a high-level logic diagram of the pulse detector 312 of the present invention. Referring now to Figs. 6, 7, and 8, signals 610 and 632 are received from the register/demultiplexer 207 of Fig. 2 through digital data signal 230. As discussed above with respect to Fig. 2, the analog section 126 sends two samples at a time to the digital section. Signal 610, also designated y_n, is the digitized value for the signal level of the most recent sample. Signal 632, also designated y_{n-1}, is the digitized value for the signal level of the next prior to the most recent sample. In the preferred embodiment, these two signals are each six bit digital values stored in two's complement number representation, with one of the bits being the sign bit. Signal 610 is fed to a multiplexer 608 and signal 632 is fed to a second multiplexer 640. These multiplexers are controlled by an exclusive-OR gate 606 which has two signals 602 and 604 as its inputs. Signal 602 determines whether the pulse detector is in acquisition or tracking mode, as described above, and signal 604 determines whether the pulse detector is in center or side sampling mode, as discussed above. When the pulse detector is in acquisition and side sampling mode, or the pulse detector is in tracking and center sampling mode, signals 610 and 632 will be selected by the multiplexers 608 and 640 respectively. When the pulse detector is in acquisition mode center sampling or in tracking mode side sampling, the multiplexers 608 and 640 will select signals 631 and 639 respectively.

Signal 631 is called a (1+D)/2 signal and is formed by adding signal 610 to signal 632 in summing junction 628 and then dividing this sum by two with a divide by two circuit 630 to produce signal 631. The summing junction 628 is a six bit add circuit and the divide by two circuit 630 is a shift. The signal 631 represents the sum of the y_n signal and y_{n-1} divided by two, therefore, it is the average of y_n and y_{n-1}.

Circuits 634, 636, and 638 produce a (1+D)/2 signal for the previous samples. That is, delay circuit 634, which is a set of D flip flops, is used to delay the y_n signal 610 by one clock cycle (two sample times), so it becomes signal y_{n-2}, which is sent to the summing junction 636. Summing junction 636 sums y_{n-2} with y_{n-1} and divides the result by two to produce the average at signal 639. As disclosed in Table 1, these (1+D)/2 signals are used in two of the four cases that the pulse detector can decode.

After passing through multiplexer 608, y_n signal 610 is sent to Figs. 7 and 8 over y_n signal 613. This signal is also delayed by delay circuit 617 to become y_{n-2} signal 614. In addition, y_n signal 613 is connected to a sign-bit circuit 616 which extracts only the sign-bit from the six-bit digital value. This sign-bit is connected to an exclusive-OR circuit 618 along with the other bits of the digital value 613. Also, the sign-bit 616 is connected to another summing junction 620 along with the output of the exclusive-OR circuit 618 and a threshold value 656. The circuits 616, 618, and 620 convert the value of y_n to an absolute value, that is, y_n is negated if it was originally a negative value to produce a positive value and is unchanged if it was originally a positive value. The threshold signal 656 is then subtracted from the absolute value to produce a non-negative value if the absolute value of y_n is greater than or equal to the threshold. The summing junction 620 produces a negative value if the absolute value of y_n is less than the threshold 656.

The output of the summing junction 622 is passed to another sign-bit circuit 624 which extracts the sign bit from the result and inverts it. This will indicate whether the absolute value of y_n is greater than or equal to the threshold value 656. The inverted sign-bit is then sent through a delay circuit 626. The output signal 627 of the delay circuit 626 has a value of one if the absolute value of y_n delayed by one clock cycle, that is, $|y_{n-2}|$, is greater than or equal to the threshold value, and is zero if it is less than the threshold.

In a similar manner, y_{n-1} signal 642 is sent to a sign-bit circuit 644, an exclusive-OR circuit 646, and a summing junction 648. The output of the summing junction 648 indicates whether the absolute value of the y_{n-1} signal is greater than or equal to the threshold value 656. This result is passed through a sign-bit circuit 650 and then inverted to indicate whether the absolute value of y_{n-1} is greater than or equal to the threshold value 656.

Fig. 7 shows a high-level logic diagram of the section of the pulse detector 312 that detects pulses during data tracking. Referring now to Fig. 7, comparator circuits 702 and 704 compare the results of the last three samples taken of the signal. Y_n signal 613 is the most recent sample, y_{n-1} signal 642 is the previous sample, and y_{n-2} signal 614 is the sample prior to signal 642. The outputs of the comparator 702 and 704 are connected to a number of AND gates and delay circuits to implement the tracking equations shown in Table 1.

AND gate 706 determines whether a positive pulse is detected at time n (with the peak at time n-1). Output 736 of comparator 702 is a logical one if y_{n-2} is less than y_{n-1}, and output 740 of comparator 704 is a logical one if y_n is less than y_{n-1}. These signals are ANDed together with the inverted sign of the signal level of y_{n-1} to produce a logical one from AND gate 706 if a positive pulse has been detected at time n.

AND gate 708 determines whether there is a negative pulse at time n (with the peak at time n-1). Output 738 of comparator 702 is a logical one if y_{n-1} is less than y_{n-2}, and output 742 of comparator 704 is a logical one if y_{n-1} is less than y_n. These signals are ANDed together with the sign of y_{n-1} to produce a logical one from AND gate 708 if a negative pulse is detected at time n.

AND gates 716 and 718 determine whether a pulse is detected at time n-1 (with the peak at time n-2). Output 736 of comparator 702 indicates that y_{n-2} is less than y_{n-1}. Output 740 of comparator 704, after being delayed by delay circuit 710, indicates that y_{n-2} is less than y_{n-3}. Signal 744 indicates that y_{n-2} is negative, therefore, the ANDing of signals 736, 748, and 744 indicate there is a negative pulse detected at time n-1.

Signal 738 indicates that y_{n-2} is greater than y_{n-1}. Signal 746, the output of delay circuit 712, indicates that y_{n-2} is greater than y_{n-3}. By ANDing signals 738, 746, and the inverted signal 744, AND gate 718 indicates that a positive pulse is detected at time n-1.

The output of OR gate 714 indicates that a pulse, either positive or negative, is detected at time n, and the output of OR gate 724 indicates that a positive or negative pulse is detected at time n-1. AND gate 720 determines whether the magnitude of the pulse at time n was greater than the threshold value and it also determines whether the pulse detector is in tracking mode. AND gate 726 determines whether the pulse at time n-1 was greater than the threshold and whether the detector is in tracking mode. Because of the recording code used in writing data to the storage media in this application, a pulse can never occur at time n and also at time n-1. Therefore, the outputs of and gates 720 and 726 are cross-coupled to AND gates 728 and 730 to prevent detection of pulses in both locations at once. Thus, the output of gates 728 and 730 will indicate that a pulse occurred either at time n or at time n-1 but not both. These outputs are connected to the circuit of Fig. 8.

Fig. 8 shows a high level logic diagram of the portion of the pulse detector that implements the acquisition equations shown in Table 1 for acquisition mode. Referring now to Fig. 8, signal 602 indicates whether or not the detector is in acquisition mode. Sign-bit extractor circuit 802 and sign-bit extractor circuit 804 extract the sign bits from the values y_n and y_{n-1} respectively. These two sign bits are exclusive NORed in XNOR circuit 808, whose output indicates whether the sign of y_n is equal to the sign of y_{n-1}. Circuit 810 indicates whether the sign of y_{n-1} is not equal to the sign of y_{n-2}. If both these conditions are true, the output of AND gate 816 will indicate that a pulse is detected at time n (with the peak between time n and time n-1) in acquisition mode. This output is ORed with output 732 from Fig. 7 which indicates whether a pulse is detected at time n in tracking mode, therefore, signal 824 indicates whether a pulse is detected in one of the modes at time n.

The sign-bit output of circuit 804 is delayed one clock cycle to indicate the sign of signal y_{n-3}. This signal is exclusive ORed with the sign bit of y_{n-2} by circuit 814, whose output indicates whether the sign of y_{n-2} is not equal to the sign of y_{n-3}. The output of exclusive-OR circuit 810, after being inverted at the input of AND gate 818, indicates whether the sign of y_{n-1} is equal to the sign of y_{n-2}. Therefore, the output of AND gate 818 indicates whether a pulse is detected in acquisition mode at time n-1 (with the peak between time n-1 and time n-2). This output is ORed with the output 734 of Fig. 7, which indicates whether a pulse is detected in tracking mode at time n-1. Therefore, output 826 indicates whether a pulse is detected at time n-1, either in acquisition or tracking mode.

Figs. 9 through 11 show a circuit that implements the equations of Table 2. This circuit, and the equations of Table 2, provide an implementation of the pulse detector that allows the preamble section of the data to contain different types of patterns. For example, the circuit of Figs. 6-8 works well for a preamble having a 2T pattern, which is an alternating 1-0 pattern (10101010...), where a 1 indicates presence of a pulse, a zero indicates absence of a pulse, and pulses alternate in polarity. The circuit of Figs 9-11 allows a 2T pattern, a 3T pattern (100100...), and an 4T pattern (10001000...). Thus, the circuit of Figs. 9-11 and equations of Table 2 are the preferred embodiment and the best mode.

In all cases except side sampling acquisition mode, the equations of both Tables 1 and 2 detect pulses with the same latency. For side sampling acquisition, the equations of Table 1 detect peaks one sample sooner that the equations of Table 2. However, the equations of Table 2 are consistent in the amount of latency in both acquisition and tracking modes, while with the equations of Table 1 for side sampling, the latency in acquisition mode is one sample shorter than the latency in tracking mode.

The equations of Table 2 show detection of a pulse at time y_n. As with the circuits of Figs. 6-8, the circuit of Figs. 9-11 processes two signal samples simultaneously. The equations for the second sample are the same as the equations of the first sample shown in Table 2, with one time delay. Thus, for example, y_n in Table 2 is replaced by y_{n-1}, y_{n-1} is replaced by y_{n-2}, etc., for the second sample.

Referring now to Figs 9-11, Fig. 9 shows a series of comparators, 912-922, that compare the values of y_n, y_{n-1}, y_{n-2}, and y_{n-3}, and compares y_n and y_{n-1} to the threshold value. Most of the comparators have a delay circuit, 924-938, on their output to also provide a comparison of y_{n-2}, y_{n-3}, y_{n-4}, and y_{n-5}. These circuits thus provide all the inputs necessary for the equations of Table 2, and the equivalent equations for the second sample as discussed above.

Fig. 10 implements the equations of Table 2 for the current sample, and Fig. 11 implements the equations of Table 2 for the first previous sample. Therefore, in the following description, Figs. 10 and 11 will be described together.

In Figs. 10 and 11, NOR circuits 1002 and 1102 determine if the absolute value of the signal is greater than the threshold. As shown in Table 2, this comparison is done in tracking mode, but not in acquisition mode. Multiplexers 1020 and 1120 select side or center sampling mode. Multiplexers 1008 and 1108 select positive or negative pulses in center sampling mode, and multiplexers 1018 and 1118 select positive or negative pulses in side sampling mode. NAND circuits 1004, and 1104 implement the equations for center sampling positive pulses, and NAND circuits 1010 and 1110 implement the equations for center sampling negative pulses. Multiplexers 1006, 1106, 1012, and 1112 allow a user to select one of two equations during center sampling. These equations are shown in Table 2, equation 3, for positive and negative pulses of center sampling. This provides flexibility to the user of the circuit. NAND circuits 1016 and 1116 implement the equations for side sampling, positive pulses, and NAND circuits 1022 and 1122 implement the equations for side sampling, negative pulses. Condition 3 of Table 2 may be omitted during tracking mode when condition 4 is used.

Because data recorded on a magnetic storage medium is recorded using small magnets having alternating north-south poles, the signal detected by the read head will always have alternating positive and negative pulses. Therefore, for detecting signals read from a magnetic recording medium, the equations of Tables 1 and 2 could be easily modified to detect a pulse only if the pulse is opposite in polarity to the previous pulse detected. This could provide additional noise immunity.

Having thus described a presently preferred embodiment of the present invention, it will now be appreciated that the aspects of the invention have been fully achieved, and it will be understood by those skilled in the art that many changes in construction and circuitry and widely differing embodiments and applications of the invention will suggest themselves without departing from the spirit and scope of the present invention. The disclosures and the description herein are intended to be illustrative and are not in any sense limiting of the invention, more preferably defined in scope by the following claims.

Table 1

	Acquisition	Tracking	
Side Sampled	1) $SGN(y_n) = SGN(y_{n-1})$ 2) $SGN(y_{n-1}) \neq SGN(y_{n-2})$	y_n sequence is filtered by $(1+D)/2$, then fed into Center Sampled tracking pulse detector	
		Pos Pulse	Neg Pulse
Center Sampled	y_n sequence is filtered by $(1+D)/2$, then fed into Side Sampled acquisition pulse detector	1) $y_{n-1} > Thr$	1) $y_{n-1} < -Thr$
		2) $y_{n-1} > y_{n-2}$	2) $y_{n-1} < y_{n-2}$
		3) $y_{n-1} > y_n$	3) $y_{n-1} < y_n$

where Thr is the Threshold value

Table 2

	Positive Pulse	Negative Pulse
Side Sampled	1) $y_n < y_{n-2}$	1) $y_n > y_{n-2}$
	2) $y_{n-3} < y_{n-1}$	2) $y_{n-3} > y_{n-1}$
	3) $y_{n-4} < y_{n-2}$	3) $y_{n-4} > y_{n-2}$
Trk Only	4) $y_{n-1} > Thr$	4) $-y_{n-1} > Thr$
Center Sampled	1) $y_n < y_{n-1}$	1) $y_n > y_{n-1}$
	2) $y_{n-2} < y_{n-1}$	2) $y_{n-2} > y_{n-1}$
	3) $y_{n-3} < y_{n-2}$	3) $y_{n-3} > y_{n-2}$
	or	or
	$y_{n-3} < y_{n-1}$	$y_{n-3} > y_{n-1}$
Trk Only	4) $y_{n-1} > Thr$	4) $-y_{n-1} > Thr$

where Thr is the Threshold value

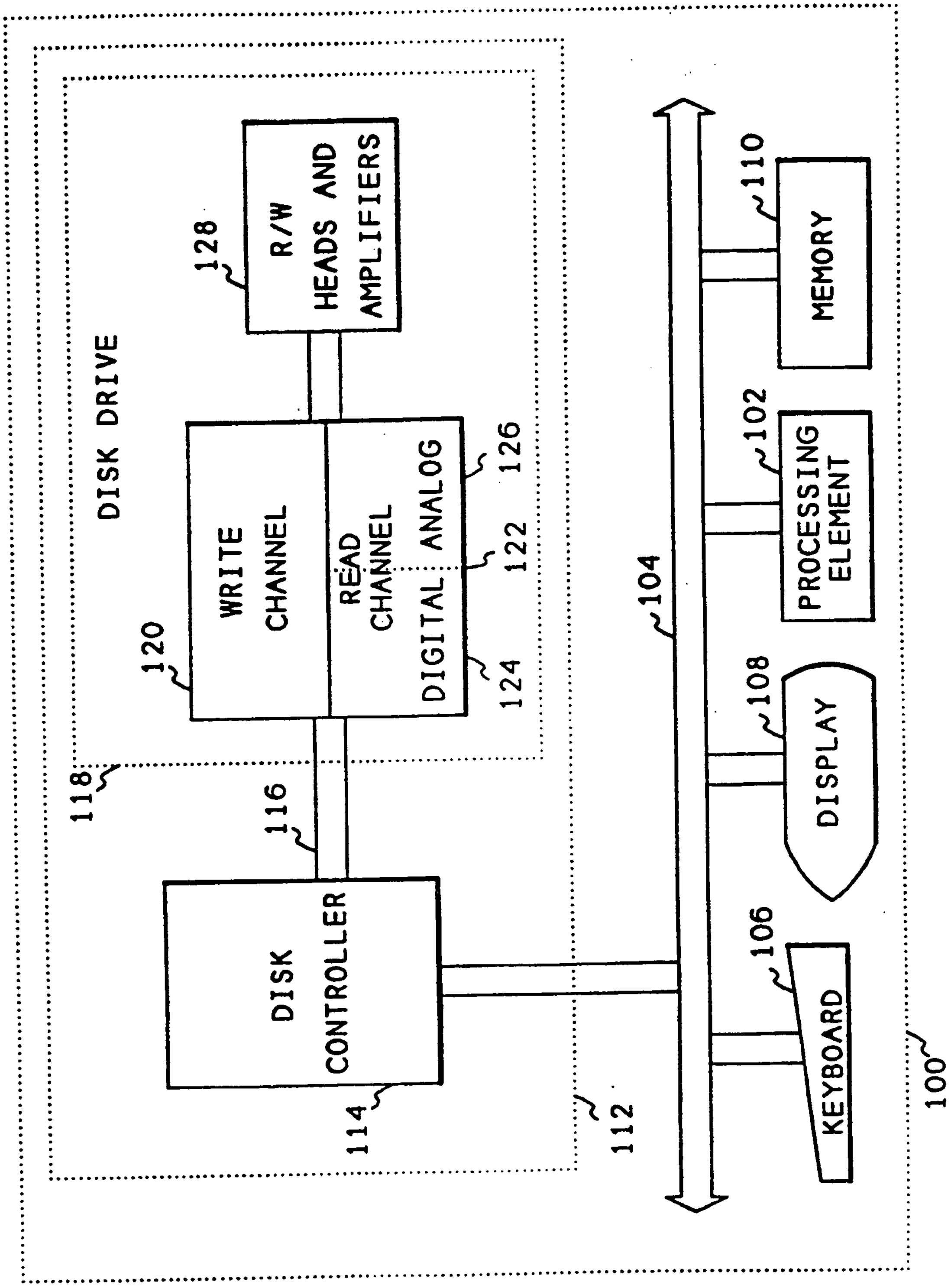

DISK DRIVE
R/W HEADS AND AMPLIFIERS
128
WRITE CHANNEL
READ CHANNEL
DIGITAL
ANALOG
120
124
122
126
118
116
DISK CONTROLLER
114
112
104
MEMORY
110
PROCESSING ELEMENT
102
DISPLAY
108
KEYBOARD
106
100

FIG. 1

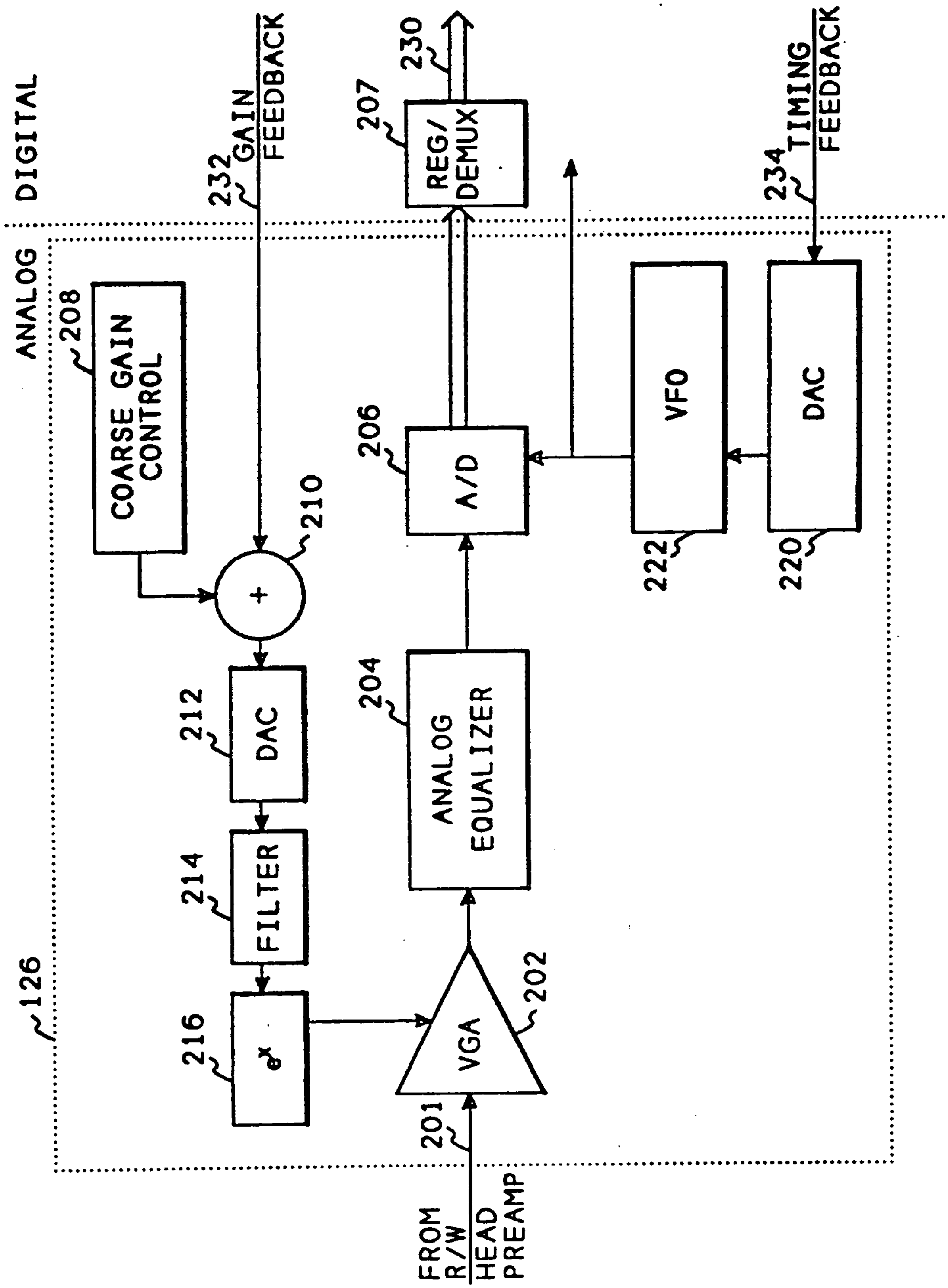

DIGITAL
ANALOG
232
GAIN FEEDBACK
207
REG/ DEMUX
230
234
TIMING FEEDBACK
208
COARSE GAIN CONTROL
210
+
206
A/D
VFO
DAC
222
220
212
DAC
214
FILTER
216
e^x
204
ANALOG EQUALIZER
202
VGA
201
126
FROM R/W HEAD PREAMP

FIG. 2

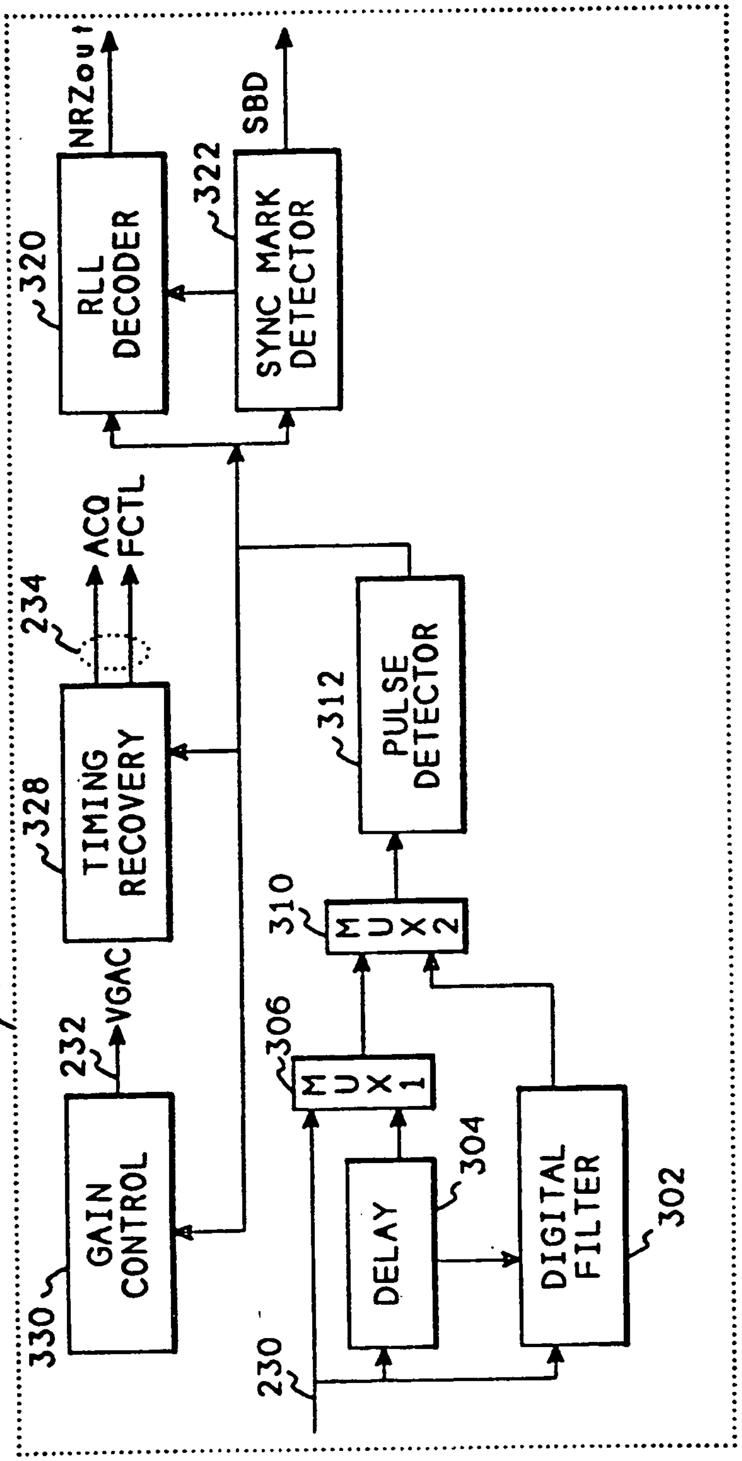

124
GAIN CONTROL
330
232
VGAC
TIMING RECOVERY
328
234
ACQ
FCTL
RLL DECODER
320
NRZout
SYNC MARK DETECTOR
322
SBD
PULSE DETECTOR
312
MUX 2
310
MUX 1
306
DELAY
304
DIGITAL FILTER
302
230

FIG. 3

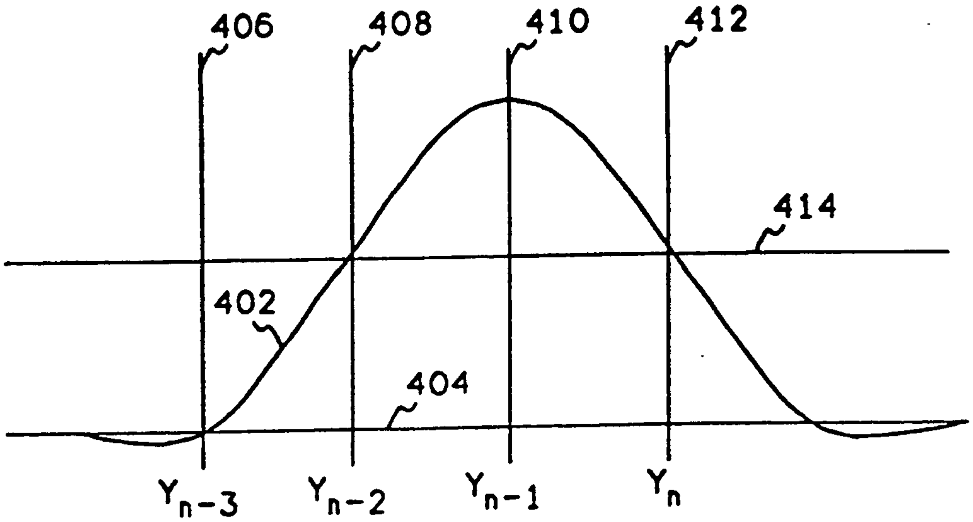

406
408
410
412
414
402
404
Y_{n-3}
Y_{n-2}
Y_{n-1}
Y_n

FIG. 4

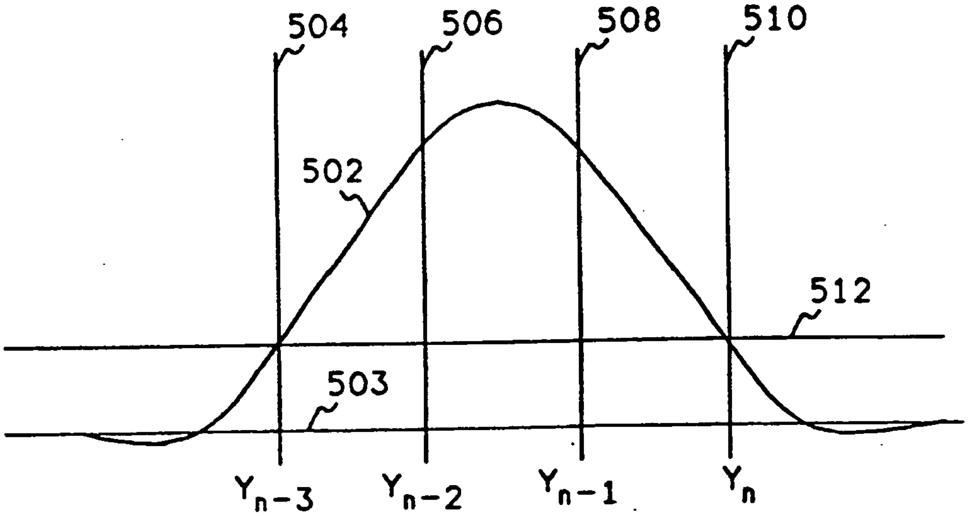

504
506
508
510
502
512
503
Y_{n-3}
Y_{n-2}
Y_{n-1}
Y_n

FIG. 5

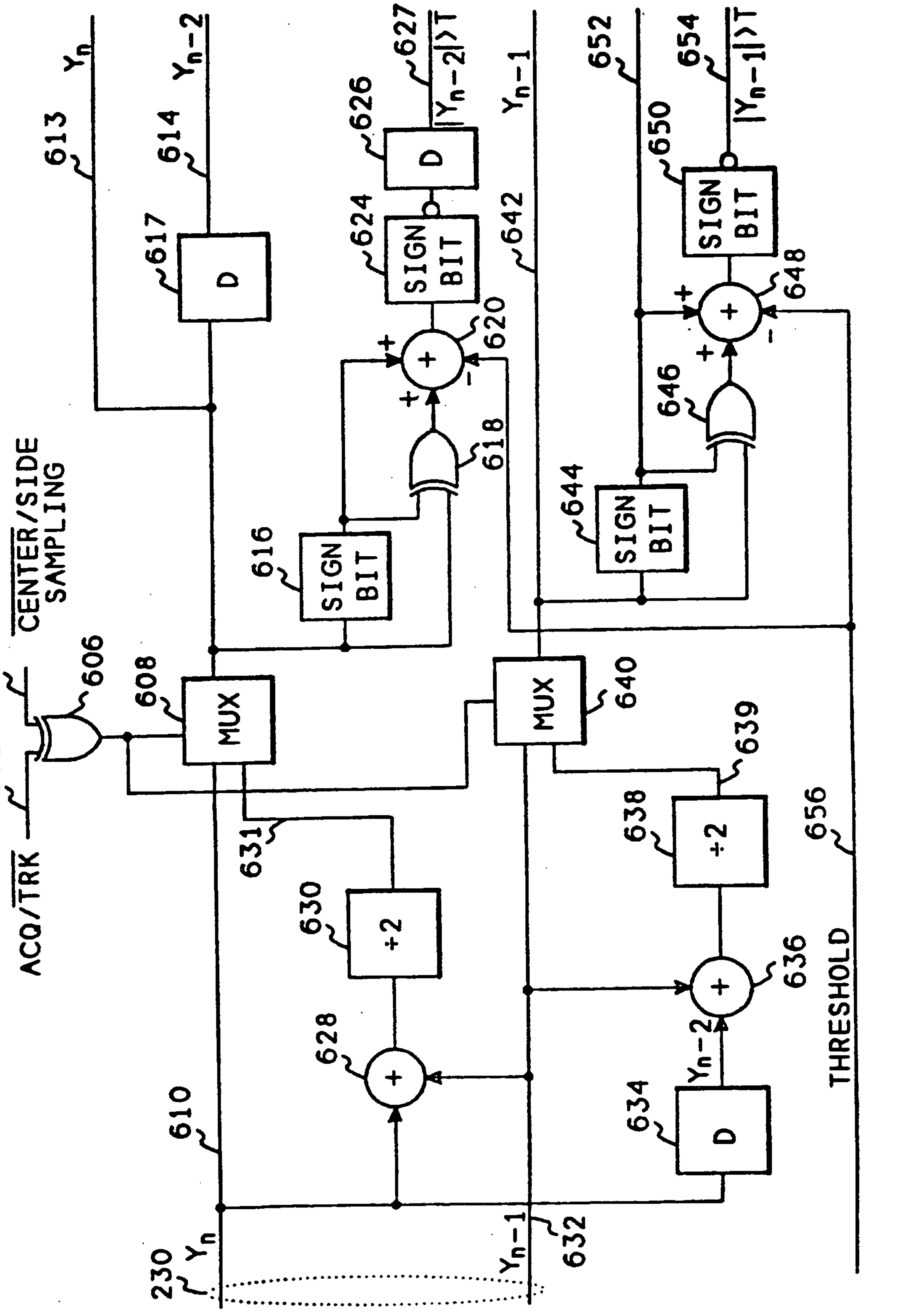

ACQ/$\overline{TRK}$
602
604
$\overline{CENTER}$/SIDE SAMPLING
606
608
MUX
610
613
614
Y_n
Y_n-2
617
D
616
SIGN BIT
618
620
+
+
−
624
SIGN BIT
626
D
627
$|Y_n-2|>T$
628
+
630
÷2
631
230
Y_n
Y_n-1
632
634
D
Y_n-2
636
+
638
÷2
639
640
MUX
642
Y_n-1
644
SIGN BIT
646
648
+
+
−
650
SIGN BIT
652
654
$|Y_n-1|>T$
656
THRESHOLD

FIG. 6

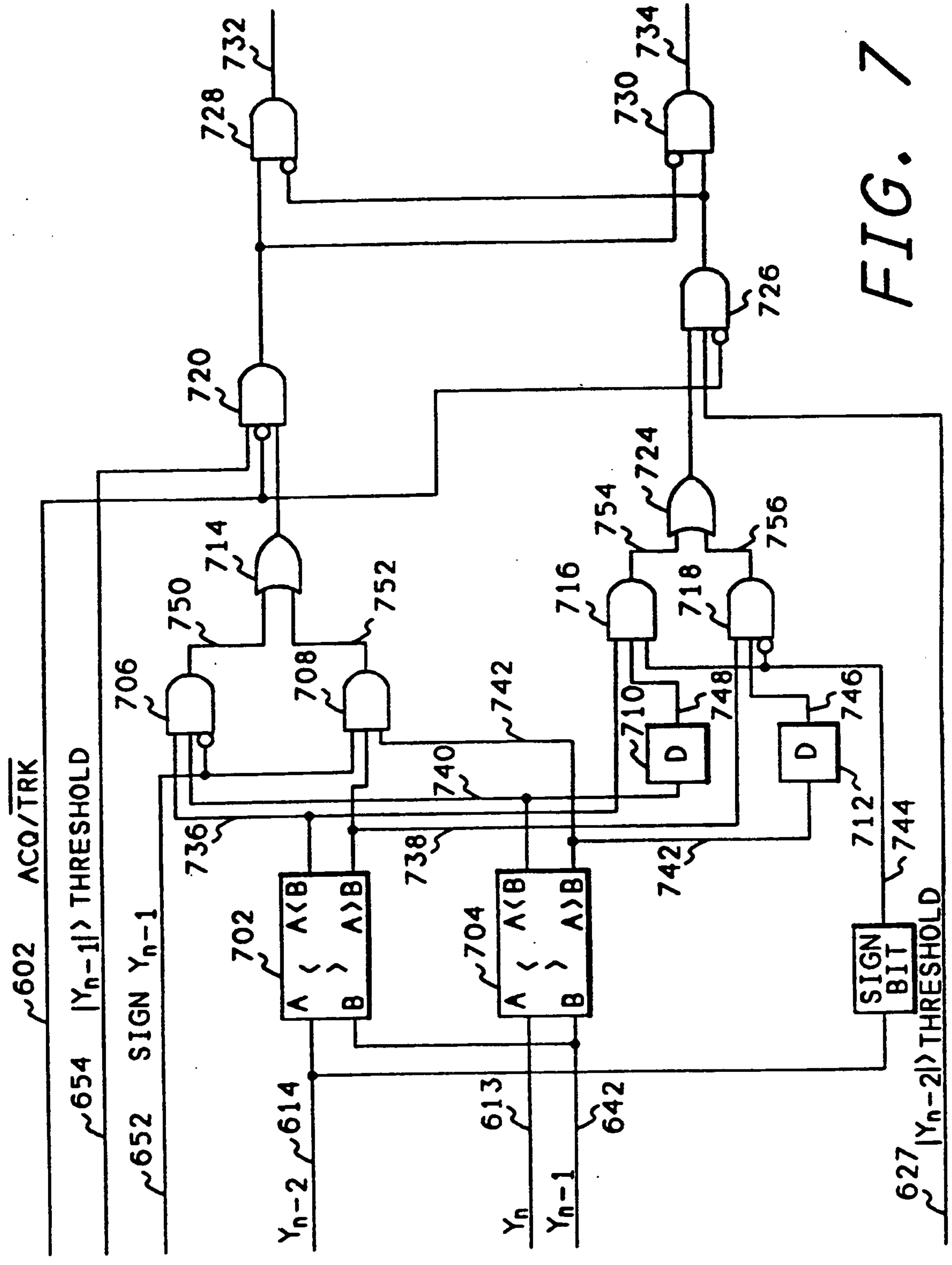

602
ACQ/$\overline{TRK}$
654
$|Y_{n-1}|$> THRESHOLD
652
SIGN Y_{n-1}
Y_{n-2}
614
Y_n
613
Y_{n-1}
642
627
$|Y_{n-2}|$> THRESHOLD
702
704
A
B
A<B
A>B
736
738
740
742
744
746
748
750
752
754
756
706
708
710
712
714
716
718
720
724
726
728
730
732
734
D
SIGN BIT THRESHOLD

FIG. 7

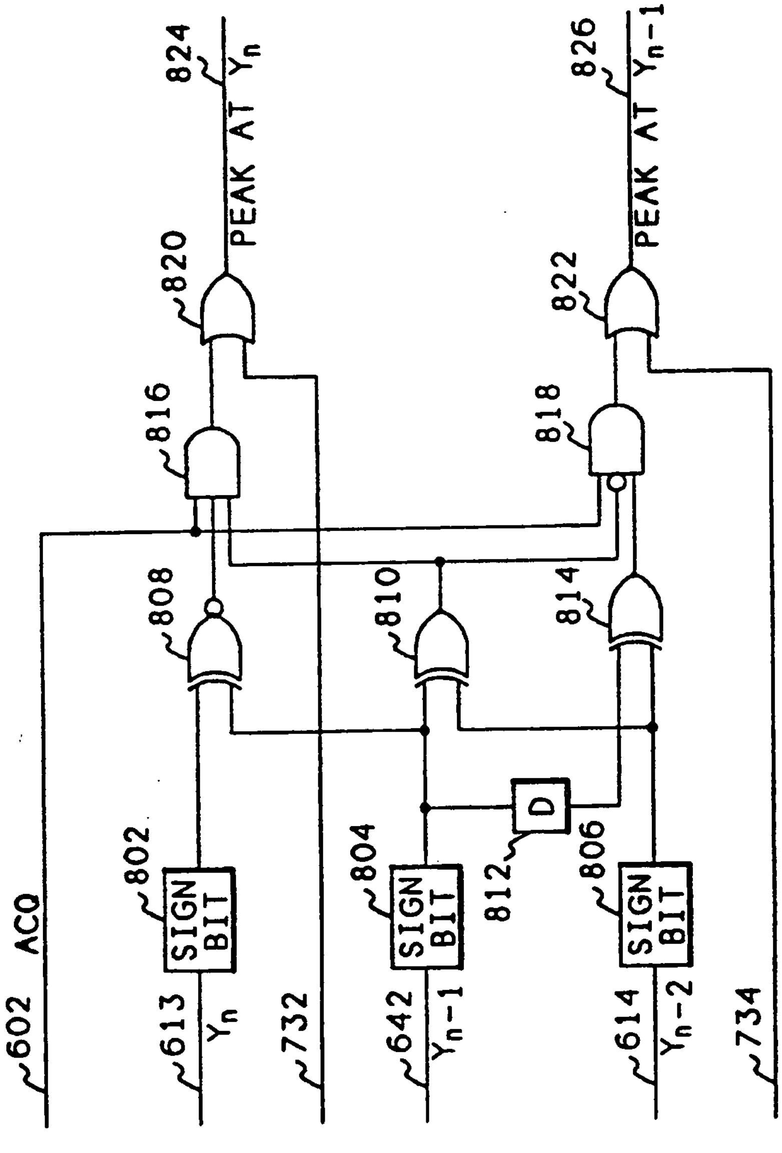
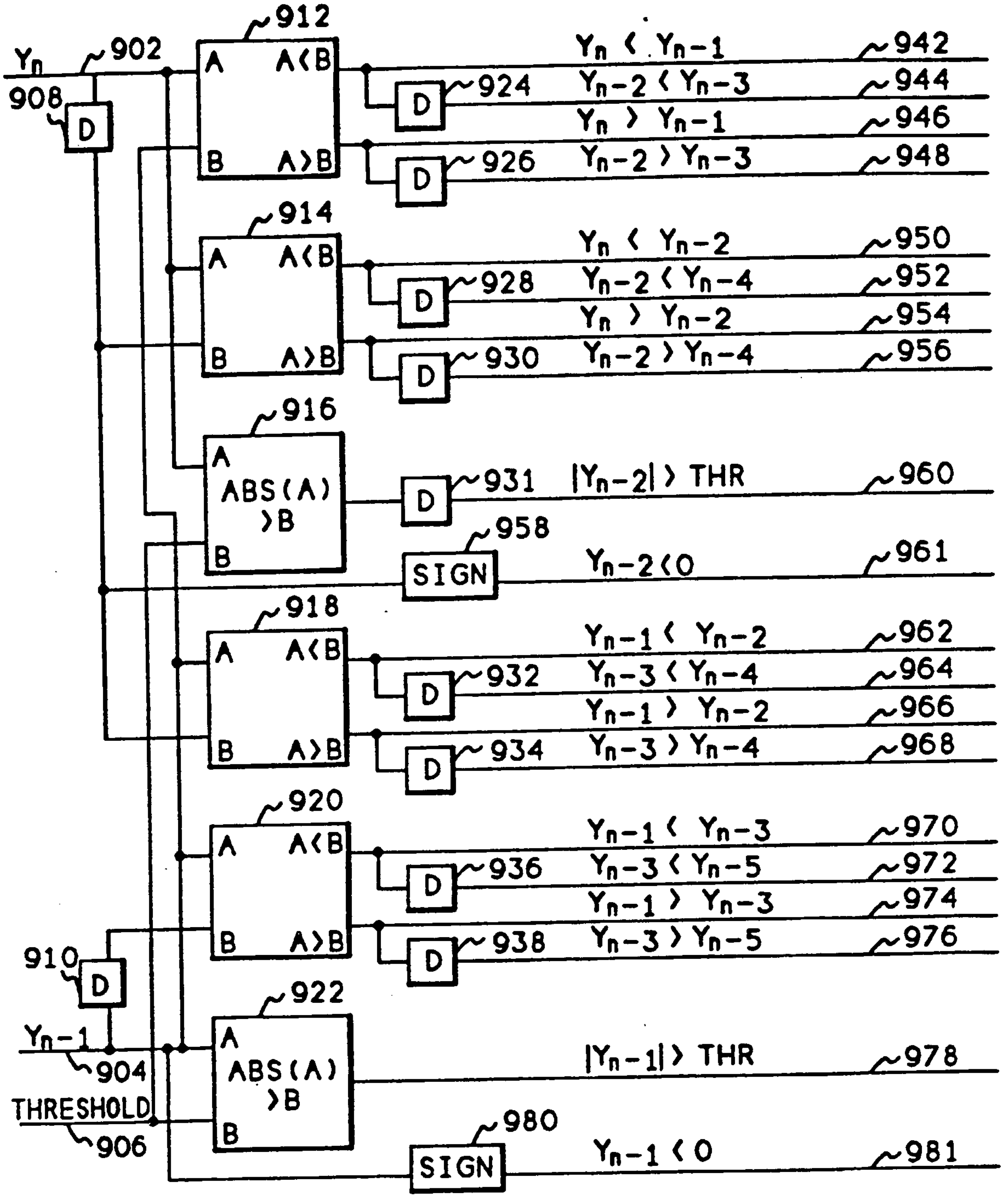

602
ACQ
613
Y_n
802
SIGN BIT
732
642
Y_{n-1}
804
SIGN BIT
812
D
614
Y_{n-2}
806
SIGN BIT
734
808
810
814
816
818
820
822
PEAK AT Y_n
824
PEAK AT Y_{n-1}
826

FIG. 8



Yn
902
908
D
912
A
A<B
B
A>B
924
D
926
D
Yn < Yn-1
942
Yn-2 < Yn-3
944
Yn > Yn-1
946
Yn-2 > Yn-3
948
914
A
A<B
B
A>B
928
D
930
D
Yn < Yn-2
950
Yn-2 < Yn-4
952
Yn > Yn-2
954
Yn-2 > Yn-4
956
916
A
ABS(A) >B
B
931
D
|Yn-2| > THR
960
958
SIGN
Yn-2<0
961
918
A
A<B
B
A>B
932
D
934
D
Yn-1 < Yn-2
962
Yn-3 < Yn-4
964
Yn-1 > Yn-2
966
Yn-3 > Yn-4
968
920
A
A<B
B
A>B
936
D
938
D
Yn-1 < Yn-3
970
Yn-3 < Yn-5
972
Yn-1 > Yn-3
974
Yn-3 > Yn-5
976
910
D
Yn-1
904
THRESHOLD
906
922
A
ABS(A) >B
B
|Yn-1| > THR
978
980
SIGN
Yn-1 < 0
981

FIG. 9

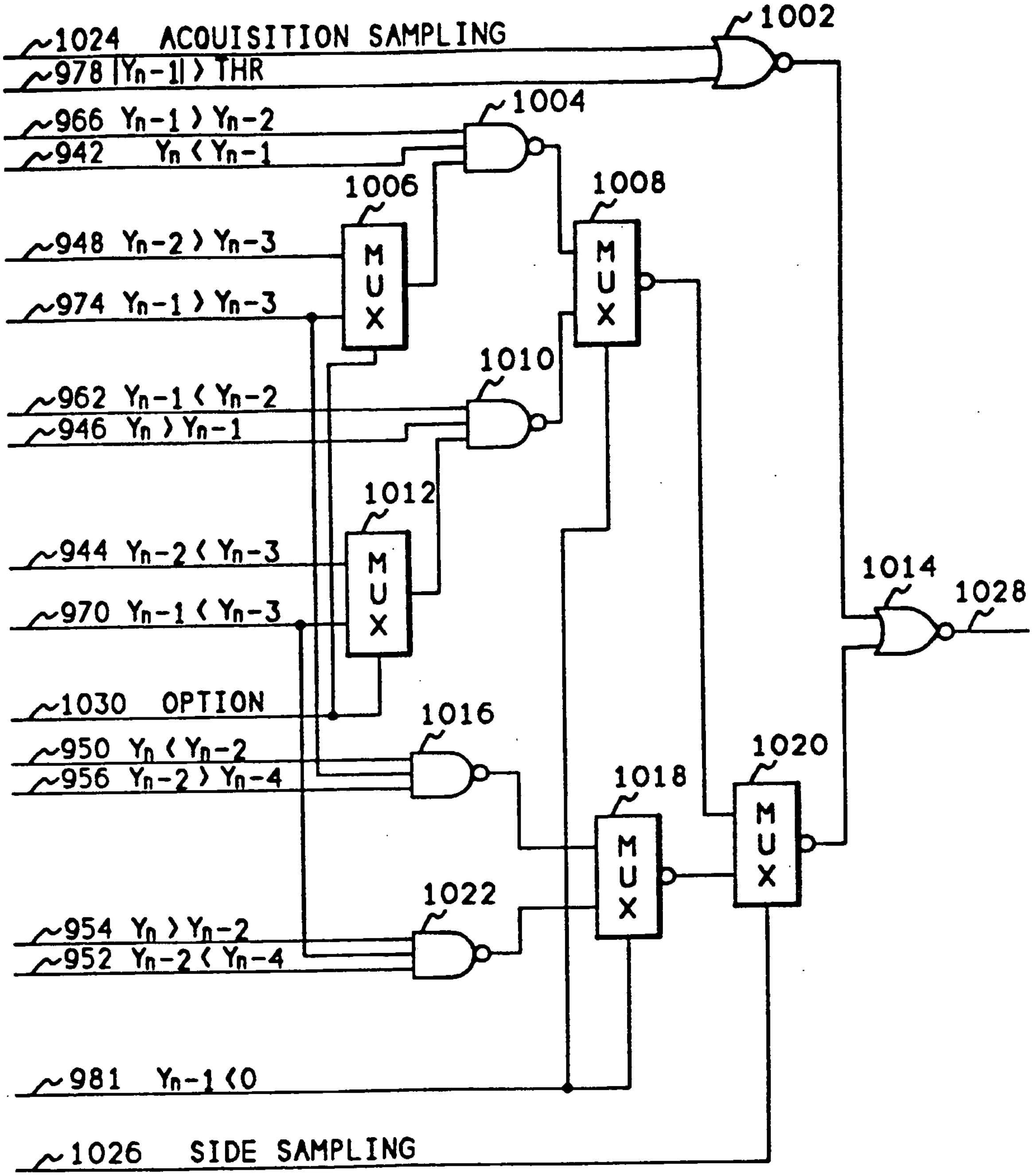
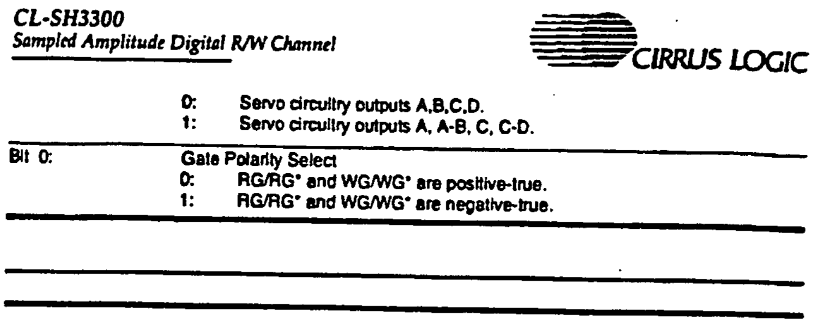

1024 ACQUISITION SAMPLING
978 |Yn-1| > THR
966 Yn-1 > Yn-2
942 Yn < Yn-1
948 Yn-2 > Yn-3
974 Yn-1 > Yn-3
962 Yn-1 < Yn-2
946 Yn > Yn-1
944 Yn-2 < Yn-3
970 Yn-1 < Yn-3
1030 OPTION
950 Yn < Yn-2
956 Yn-2 > Yn-4
954 Yn > Yn-2
952 Yn-2 < Yn-4
981 Yn-1 < 0
1026 SIDE SAMPLING
1002
1004
1006
1008
1010
1012
1014
1016
1018
1020
1022
1028
MUX
MUX
MUX
MUX
MUX

FIG. 10



	0:	Servo circuitry outputs A,B,C,D.
	1:	Servo circuitry outputs A, A-B, C, C-D.
Bit 0:	Gate Polarity Select	
	0:	RG/RG* and WG/WG* are positive-true.
	1:	RG/RG* and WG/WG* are negative-true.

applications, including data storage on optical media and data communications via modem, network, radio, or fiber-optic link.

To recover the written or transmitted data sequence, the receiver requires a clock signal synchronized with the received waveform. At each tick of this synchronized clock signal the receiver or read circuitry generates one bit of the NRZI data sequence by processing the surrounding waveform. It is often impossible or at least undesirable to store or transmit a separate synchronized clock signal with the data waveform. Instead, constraints are applied to the encoded NRZI data sequence to ensure that timing information may be extracted from the data waveform itself and used to "recover" a synchronized clock signal. Such a system is referred to as "self clocking".

Prior art magnetic disk drives typically use a data detection system known as peak detection, in which analog circuitry determines the time instant of each peak in the waveform. Peaks that meet a qualification requirement, such as exceeding a threshold amplitude, are considered to have been caused by a magnetic transition and are therefore associated with an NRZI "one" bit. The recovered clock divides the waveform into bit cells or windows where the nominal position of a peak is at the center of a window. A "one" bit is output for each window that contains a qualified peak and a "zero" bit is output for each other window. The phase and frequency of the recovered clock are adjusted in a phase locked loop so as to keep the windows properly aligned with the peaks.

Inter-symbol interference (ISI) occurs when pulses are placed so close together in a waveform that they overlap significantly. The position of a peak in a waveform can be shifted by ISI from other nearby pulses. Thus the recording density of a peak detection channel is limited by the fact that ISI must not be permitted to move a peak from one window to another. Peak shift due to ISI also causes apparent timing errors that affect the performance of the clock recovery circuit.

To achieve higher recording densities and/or greater noise immunity than peak detection channels, another class of detection methods is based on amplitude sampling of the signal waveform. Many such sampling detection methods can be implemented using either analog or digital signal processing devices. In either case, a synchronized clock signal is required to control the sample timing and a means must be provided to recover such a clock signal from the data waveform. Thus there is a need for new timing recovery methods suitable for use with sampled amplitude detectors at high recording densities. The present invention meets this and other needs.

This Application is related to Application Serial Number 07/852,015, filed March 16, 1992, of Richard T. Behrens, Kent D. Anderson and Neal Glover, entitled "Method and Apparatus for Reduced-Complexity Viterbi-Type Sequence Detectors", and Application Serial Number 07/879,938, filed May 8, 1992, of Richard T. Behrens, Trent Dudley, and Neal Glover, entitled "Digital Pulse Detector", both owned by the same entity. Each of these applications is incorporated herein by reference for all that is disclosed and taught therein.

SUMMARY OF THE INVENTION

It is an aspect of the present invention to detect signal pulses caused by bit transitions within data being read from a data storage device or data communications device.

It is an aspect of the invention to measure a phase error between each pulse in a signal waveform and a clock signal used to sample the waveform.

It is another aspect of the invention to compensate the phase error measurement for asymmetry of the pulse being detected and to compensate for nearby pulses.

It is another aspect of the invention to adjust the frequency, and thereby the phase of the clock signal so as to drive the phase error measurement toward zero.

It is another aspect of the invention to measure a frequency error between a fixed frequency pulse train and a clock signal used to sample the pulse train and adjust the frequency of the clock signal so as to drive the frequency error measurement toward zero.

The above and other aspects of the invention are accomplished in the preferred embodiment in a mixed analog and digital timing circuit for timing the conversion of an analog signal to a digital signal. The circuit has an analog to digital converter for converting the analog signal to digital sample values at controlled sampling times, and a variable frequency oscillator for controlling when the analog to digital converter performs the conversion. A pulse detector detects pulses in the digitized signal. A phase error circuit subtracts one of two sample values for each pulse from the other to create a digital phase error measurement value. The two sample values occur on either side of the peak of the pulse detected by the pulse detector. The digital phase error measurement value is filtered using a digital filter whose output is connected to the variable frequency oscillator to adjust the sample timing to cause samples to be taken at the desired time instants.

The system also contains a set point register that contains a value used to adjust the desired sample phase. The set point may be used to compensate for the pulse being asymmetrical. This value is always added to the digital phase error measurement value.

In addition to the set point register, the system contains compensation registers for pulses that occur at two sample times on either side of the detected pulse, since these pulses are close enough to the detected pulse to change its shape. These values are only added into the digital phase error value if a pulse is also detected at these sample times. A third compensation register is also present for pulses that occur three sample times after the detected pulse, and the value of this register is added to the digital phase error value if this pulse is detected.

The system also contains a lock to reference mode that includes synthesizing a fixed frequency data pattern, inserting the data pattern in place of the read signal and locking the variable frequency oscillator to the synthesized frequency.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other aspects, features, and advantages of the invention will be better understood by reading the following more particular description of the invention, presented in conjunction with the following drawings, wherein:

Fig. 1 shows a block diagram of the environment of the invention and illustrates the read channel that contains the invention;

Fig. 2 shows a block diagram of the analog and analog to digital conversion circuitry of the read channel containing the invention;

Fig. 3 shows a block diagram of the digital circuitry of the read channel containing the invention;

Fig. 4 shows a signal waveform and illustrates center sampling of the pulses;

Fig. 5 shows a signal waveform and illustrates side sampling of the pulses;

Figs. 6 and 7 show signal waveforms to illustrate overlapping pulses;

Fig. 8 shows a block diagram of the timing recovery circuit of Fig. 3;

Figs 9-10 show a logic block diagram of the acquisition circuit of the phase and frequency detector of Fig. 8;

Figs. 11-12 show a logic block diagram of the tracking circuit of the phase and frequency detector of Fig. 8; and

Fig. 13 shows a diagram of the loop filter circuit of the timing recovery block of Fig. 3.

DESCRIPTION OF THE PREFERRED EMBODIMENT

The following description is of the best presently contemplated mode of carrying out the present invention. This description is not to be taken in a limiting sense but is made merely for the purpose of describing the general principles of the invention. The scope of the invention should be determined by referencing the appended claims.

Fig. 1 shows a block diagram of a typical environment of the invention. Referring now to Fig. 1, a computer system 100 contains a processing element 102 which communicates to other elements of the computer system 100 over a system bus 104. A keyboard 106 and a display 108 allow a user of the computer system 100 to communicate with the computer system 100. A memory 110 contains programs and data which cause the computer system 100 to perform operations desired by the user.

A disk data storage system 112 is connected to the system bus 104 for storing data and programs within the computer system 100. A disk controller 114 within the disk device 112 communicates to the system bus 104 and controls the operations of a disk drive 118, possibly in conjunction with a local microprocessor (not shown) within the disk data storage system 112. The disk drive 118 performs the storage function, typically storing the data on magnetic media. A bus 116 connects the disk controller 114 to the disk drive 118, specifically connecting to a write channel 120 to write data onto the disk through write heads and amplifiers 128. When data is being read from the disk through the read head and amplifiers 128 the data comes back through the read channel 122 which contains the timing recovery circuit of the present invention. The read and write heads may be physically the same heads. The data first passes through the analog section 126 of the read channel 122 and then through the digital section 124 of the read channel 122 before being sent on the bus 116 to the disk controller 114. After being processed by the disk controller 114, the data is then sent over the system bus 104 to the memory 110 and/or the processing element 102.

The disk controller 114 also connects to other circuits, not shown, within the disk drive 118, such as a circuit which moves the read/write heads over the surface of the data storage media.

Although not shown in Fig. 1, the digital timing circuit of the present invention can be used for synchronizing a clock to pulses within data received from a transmission line, such as a telephone line or local area network, in a data communications receiver. It may also be used in any other device that must time and detect pulses within a signal.

Fig. 2 shows a block diagram of the analog circuitry 126 of the read channel 122. Referring now to Fig. 2, when a read head is passing over a track of the data storage medium, it picks up a signal which is amplified by a preamplifier, not shown. After this preamplification, the signal 201 is passed to a variable gain amplifier 202. The signal is further amplified by the variable gain amplifier 202 and passed through an analog multiplexer 203 and then to an analog equalizer circuit 204, which filters the signal as desired, for example, so as to remove unwanted high frequencies and shape the remaining spectrum, to an Analog to Digital converter 206. The A to D converter 206 converts the analog signal into a sequence of digital values, providing six bits of digital information per sample in the preferred embodiment, and then the data is passed to a register/de-multiplexer 207. In the preferred embodiment, the digital section 124 of the read channel 122 processes two samples in parallel. To create these two samples, the register/de-multiplexer 207 stores every other sample taken by the A to D converter 206. After the second sample is taken, the data from the two samples is passed to the data bus 230. The bus 230 is clocked by a single half-frequency clock signal.

The timing necessary for converting the data, also called taking a sample, in the A to D converter 206 is supplied by a variable frequency oscillator 222 which is controlled by the output of a digital to analog converter (DAC) 220. The input to the DAC 220 comes from the digital section of the read channel 124 as timing feedback signal 234. Alternatively, the DAC 220 may be an integral part of the variable frequency oscillator in such a way that the digital input controls the frequency directly.

A reference signal 236 is created by a frequency synthesizer 240 running at the frequency used for writing data, and dividing this frequency by four in the divide by four circuit 238. The output of the divide by four circuit 238 is a square wave with period 4T which is connected to the analog multiplexer 203. The filter 204 passes the fundamental frequency of the square wave but rejects the higher harmonics, thus converting the reference signal into a sinusoid with frequency 1/4T. This sinusoidal reference signal is equivalent to the data signal typically produced by alternating transitions and spaces, i.e. 101010... i~~f~~n NRZI representation. It is often called a "2T" pattern because the distance between transitions is 2 bit cells. The ~~reference~~ 2T pattern is a pulse pattern often used for a preamble signal at the start of a data record. The reference pattern signal 236 is used to set the VFO 222 to a nominal frequency before locking to data preamble.

R.B. 9-2
TDD 9-2
JG 9-3
R.B. 9-2
TDD 9-3
JG 9-3

The gain of the variable gain amplifier 202 is controlled through a gain feedback signal 232 which originates in the digital portion 124 of the read channel 122. The gain feedback signal 232 is input to a summing junction 210 which has a coarse gain control value as its other input. The coarse gain control can be set by the disk controller 114, or a local microprocessor (not shown) within the disk drive, to provide a nominal gain level which is then adjusted up or down by the gain feedback signal 232. After being summed with the coarse gain control value, the feedback signal is sent to a digital to analog converter 212 and then to a filter 214. Because of the nature of many digital to analog converters, the output of the DAC 212 may contain glitches when it is changing values. Therefore, the filter 214 may be necessary to remove these glitches in the feedback signal. After being filtered, the signal is exponentiated by the exponential converter block 216 and then connected to the variable gain amplifier 202. This conversion makes the small-signal gain control dynamics independent of the input signal amplitude.

Fig. 3 shows a block diagram of the digital section 124 of the read channel 122 (Fig. 1). Referring now to Fig. 3, the digital data signal 230 from Fig. 2 is input to a delay circuit 304 and a digital filter circuit 302. The outputs of these two circuits are connected to multiplexers 306 and 316, whose outputs are connected to multiplexers 310 and 318 respectively. Multiplexers 310 and 318 also receive signal 230. Multiplexers 306 and 316 may be selected independently, however, multiplexers 310 and 318 are always selected together. In this manner, either the digital data signal 230 or the filtered/delayed signals may be selected for input to both a pulse detector 312 and a timing recovery circuit 328. If the filter signal is selected for either the pulse detector 312 or the timing recovery 328, either the filter signal or the delay signal must be selected for input to the other of these circuits. The filter 302 inserts a delay into the digital data signal 230, so if one circuit selects the filtered signal, the other circuit input must be delay compensated by selecting the delayed signal from circuit 304 or by also selecting the filter signal from filter 302.

One circuit suitable for use as the pulse detector 312 is disclosed in U.S. Patent Application Serial Number 07/879,938.

The output of the pulse detector 312 is connected to a gain control circuit 330 which provides the gain feedback signal 232 that connects to Fig. 2. The output of the pulse detector 312 is also connected to the timing recovery circuit 328 whose output 234 connects to the digital to analog converter 220 of Fig. 2. The output of the pulse detector 312 may also be connected to a sync mark detector 322 and an RLL decoder 320, as shown in Fig. 3, or a more sophisticated data detector (not shown) may be connected to the sync mark detector 322 and the RLL decoder 320. The output of the RLL decoder 320 and the sync mark detector 322 are connected to the disk controller 114 (Fig. 1) through the bus 116.

The pulse detector 312 and the timing recovery circuit 328 are designed to process pulses using one of two types of sampling methods, selected by the user. The first sampling method is called center sampling wherein one of the samples taken will arrive at or near the center, or peak, of a pulse as shown in Fig 4. The location of the sampling is controlled by the timing recovery block 328. In the other method of sampling, called side sampling, the timing recovery block 328 will adjust the timing of the VFO 222 (Fig. 2) such that two consecutive samples are taken wherein one of the two samples appears on one side of the peak of the pulse and the other of the two samples appears on the other side of the peak as shown in Fig. 5. The user of the system determines whether center sampling or side sampling is used by setting a bit in a control register through the interface 116 (Fig. 1).

For either center or side sampling, sampling can occur during two different time segments within a data record being read from the disk media. The first time segment sampling occurs is called acquisition mode, because it occurs when the gain control and timing control are acquiring the gain and timing relationships of the pulses. This occurs when the read head is passing a preamble portion of the data record which always has a known data pattern to facilitate acquisition of the timing and gain. When data is being transferred, a different mode is used for the pulse detector and timing recovery, called tracking, since data has an irregular and a priori unknown pattern of pulses.

Therefore, the pulse detector and timing recovery circuits are designed to process pulses under four separate conditions. The first condition is acquisition mode using side sampling, the second is acquisition mode using center sampling, third is tracking mode using side sampling, and fourth is tracking mode using center sampling.

The timing recovery circuit analyzes the current sample of the amplitude of the signal as well as the previous two samples of the signal amplitude. Using these three samples, Table 1 shows equations for the timing recovery circuit 328.

In Table 1, n is the time of the current sample, n-1 is the first previous sample, n-2 is the second previous sample and n-3 is the third previous sample. y_x is the sample value for a sample taken at time x. SGN is the arithmetic sign of the sample value. P_x has a value of 1 if a pulse is detected during sample x, and P_x has a value of 0 is a pulse is not detected during sample x. t_a is the acquisition timing set point, and t_t is the tracking timing set point. a, b, and c are constants used to adjust for nearby pulses. The equations of Table 1 will be further described below. Also, the equations of Table 1 assume pulse detection at time n of a pulse whose peak occurs at time y_{n-1} for center sampling, or between

y_{n-1} and y_{n-2} for side sampling. The circuits of Figs. 9-12 processes two signal samples simultaneously. The equations for the second sample are the same as the equations of Table 1, with one time delay. Thus, for example, y_n in Table 1 would be replaced by y_{n-1}, y_{n-1} would be replaced by y_{n-2}, and y_{n-2} would be replaced by y_{n-3}.

Fig. 4 shows a signal waveform of an isolated pulse and illustrates center sampling of the pulse. Referring now to Fig. 4, a signal waveform 402 is shown having four samples taken with the sample identified by reference 412, at time n, being the most recent sample. Sample 410, at time n-1, is the sample just prior to the most recent sample, sample 408 is the next previous sample and sample 406 is the oldest of the four samples shown. Negative pulses would appear as a mirror image of Fig. 4. The samples taken of the pulse are also identified as "a", "b", "1", and "c". Sample "a" occurs at time n-3 of each pulse, sample "b" occurs at time n-2 of each pulse, sample "1", occurs at time n-1 of each pulse, and sample "c" occurs at time n of each pulse. The sample labeled "1" is so labeled because the gain circuit adjusts the level of this sample to a nominal value of 1.

Fig. 5 shows a signal waveform of an isolated pulse and illustrates side sampling of the pulse. Referring now to Fig. 5, a signal waveform 502 is shown as a positive level above a baseline 503. Four samples are shown, with the most recent sample being sample 510. Sample 508 is the sample previous to the most recent, 506 is the next previous sample and sample 504 is the oldest of the four samples shown. These samples are also identified by the time references n through n-3.

Figs. 6 and 7 show waveforms and sample times for a typical pulse for both side and center sampling, and will be used to illustrate the equations shown in Table 1.

Referring now to Fig. 6, a waveform 602 is shown with side sampling mode, with the four samples of the waveforms, "a", "b", "1", and "c", being identified for the value of the waveform at sample times n-3, n-2, n-1, and n, respectively. Because of the type of data recording code being used, pulses can never occur at two successive sampling times, thus pulses must be separated by at least two sample times. Therefore, the pulse illustrated by dashed line 604 is the closest a pulse could occur prior to the pulse 602 and the pulse identified by dashed line 606 is the earliest another pulse could occur after the pulse 602.

Considering the side sample tracking equation of Table 1, if pulse 602 is a symmetric isolated pulse, that is, neither pulse 604 nor pulse 606 occurs, the phase error will be zero if the value of the sample taken at n-2, that is, the "b" sample, is equal to the value of the sample taken at n-1, that is, the "1" sample. This result is shown in the side sample tracking equation as $y_{n-2} - y_{n-1}$, where y_x represents the sample value at time x. Since pulses can occur in both positive and negative directions, the sign of y_{n-1} is multiplied into this equation to account for negative sample values. That is, the equation

$$\mathrm{SGN}\ (y_{n-1}) * (y_{n-2} - y_{n-1})$$

will give the phase error for a symmetric isolated pulse. This phase error will be zero if the peak of the pulse occurs exactly at the center between the "b" and "1" samples, and will be non-zero if the peak is offset from the center. If the peak is offset left of center, the "b" sample value will be higher than the "1" sample value, resulting in a positive phase error value, and if the peak is offset right of center, the "b" sample value will be lower than the "1" sample value, resulting in a negative phase error value.

There are many factors that may cause a pulse to be asymmetric, including magnetic, geometrical, and electronic factors. Because of this possible asymmetry, the term t_t is introduced into the equation. The value for t_t will be established by calibrating the device, however, Table 2 shows nominal values for t_t for different types of data patterns used during acquisition. Therefore, the equation

$$\text{SGN}\ (y_{n-1}) * (y_{n-2} - y_{n-1}) + t_t$$

gives the phase error equation for an isolated, possibly asymmetrical, pulse.

Since pulse 604 might have occurred two sample times earlier than pulse 602, the effects of this pulse must be considered in the equations. The term "$+ c * P_{n-2}$" accounts for the effects of the pulse 604. Sample "c" of pulse 604 occurs at the same time as sample "b" of pulse 602. Therefore, if there was a peak detected at time n-2, i.e. p_{n-2} is one, the phase error is corrected by adding a constant value c to offset the "c" sample of pulse 604. This constant value is determined by calibration of the device and is input to the circuit through the interface 116 (Fig. 1).

Similarly, a pulse could occur two sample times after pulse 602, represented in Fig. 6 by pulse 606. For this pulse, the "a" sample occurs at the same time as the "1" sample of pulse 602. Therefore, the phase error is corrected by subtracting a constant "a" if there is a pulse at time n+2, i.e. if P_{n+2} is one. This constant value is also determined by calibration of the device and input to the circuit through the interface 116 (Fig. 1).

The equation for side sample acquisition mode is similar to the side sample tracking mode equation, except that the corrections for the pulses 604 and 606 need not occur. These corrections are unnecessary, because in acquisition mode the data pattern is known, since the same preamble is written before every data record. Therefore, the entire compensation for asymmetry and intersymbol interference can be accomplished with a single term t_a. Table 2 shows nominal values for t_a for the three types of acquisition patterns used in the present invention. This constant value is also determined by calibration of the device and input to the circuit through the interface 116 (Fig. 1).

Fig. 7 shows waveforms when center sampling is being used. Table 1 shows the center sampling equations, and the center sampled acquisition equation is very similar to side sampled acquisition equation, except that in center sampling the "b" and "c" samples of each pulse are used to determine phase error. Therefore, the center sampled acquisition equation substitutes y_n for the y_{n-1} in the side sampled acquisition equation.

The center sampled tracking equation incorporates the same kind of adjustments that were incorporated into the center sampled acquisition equation, but the details differ. As shown in Fig. 7, a pulse 704 can overlap the pulse 702, and if this occurs, the "c" sample of pulse 704 overlaps the "b" sample of pulse 702. Therefore, the center sampled tracking equation adds the term "+ c * P_{n-2}" to compensate for the pulse 704, where c is the same constant described for the side sampling equations. Similarly, pulse 706 can overlap pulse 702 and the center sampled tracking equation adds the term "- b * P_{n+2}" to compensate for the pulse 706, where b is a constant value determined by calibration of the device and input to the circuit through the interface 116 (Fig. 1).

In center sampling, a third pulse, 708, can also overlap the pulse 702. One possible way to compensate for the pulse 708 would be to subtract a constant "a" times P_{n+3}. However, this requires that the detector wait at least three sample times before determining the phase error for the pulse 702. As shown in Fig. 7, the value of the "a" sample for pulse 708 will typically be very small for center sampling. Because it is very small, this term of the equation can be deferred to a later clock cycle without much consequence. When deferred by three sample times, the correction "- a * P_{n+3}" may be included in the phase error equation for the next pulse (at time n+3), at which time it is included as "- a * P_{n-3}". Therefore, the center sampled tracking equation includes the term "- a * P_{n-3}" to provide compensation for pulse 708.

Fig. 8 shows a block diagram of the timing recovery circuit 328 of Fig. 3. Referring now to Fig. 8, the timing recovery circuit 328 is shown having a phase/frequency detector 802 which is connected to a loop filter 808 to provide the timing feedback signal 234. The phase/frequency detector 802 receives the output 314 of the pulse detector 312 (Fig. 3). As discussed above and shown in the equations of Table 1, this is the signal "P". The phase/frequency detector 802 also receives the data samples 320 which may have been selected from the raw data samples 230 or after being delayed by the delay 304 or filtered by the digital filter 302. In addition, the phase/frequency detector 802 receives an acquisition/tracking signal 810 which tells the phase/frequency detector 802 whether acquisition or tracking mode is being used. Also, a read/lock signal 812 tells the phase/frequency detector whether data is being read from the system, or whether the detector is in lock to reference mode.

Lock to reference mode is used to set the frequency of the VFO 222 (Fig. 2) at a nominal frequency when data is not being read and thus minimize the initial frequency error when reading of data is started. This is accomplished by creating a periodic reference signal at a precisely controlled frequency equal to one fourth of the nominal sample rate. This signal is connected through analog multiplexer 203 (Fig. 2), into the normal data path. Thus, when not reading data, this reference signal may be used to set the VFO at the nominal frequency.

Table 3 shows the equation for the frequency detection, when the phase/frequency detector 802 is acting as a frequency detector. This equation is very similar to the center sampled acquisition phase error equation, except that the sign of y_n is reversed and the t_a term is absent. If the VFO 222 is sampling at the correct frequency, every other sample will have the same magnitude, but opposite in sign. Therefore, if the VFO 222 is sampling at the correct frequency, $y_n + y_{n-2}$ will be zero. if the VFO is low in frequency, the result is negative, and if the VFO is high, the result is positive.

Figs. 9-12 show a logic block diagram of the circuit for the phase/frequency detector 802. Figs. 9-10 show the circuit for determining phase and frequency errors during acquisition mode, and Figs. 11-12 show the circuit for determining phase errors in tracking mode. Frequency errors are only determined during acquisition mode. In the preferred embodiment of the present invention, two samples are processed simultaneously. Fig. 9 shows the processing of the first sample in acquisition mode, Fig. 10 shows the processing of the second sample in acquisition mode, and Fig. 10 also combines the results of Fig. 9 and 10 for acquisition mode. Similarly, Fig. 11 shows the processing of the first sample in tracking mode, Fig. 12 shows processing of the second sample in tracking mode, and Fig. 12 also combines the results of Figs. 12 and 13 for tracking mode.

Referring now to Fig. 9, the circuits surrounded by dashed line 917 partially solves the equations for side and center sampling acquisition mode and frequency detection to produce the result of:

$$\mathrm{SGN}\ (y_{n-1}) * y_{n-2} + t_a$$

for phase error detection and

$\mathrm{SGN}\ (y_{n-1})\ *\ y_{n-2}$

for frequency detection.

Multiplexer 912 selects either y_n or the inverted value of y_n based on the sign of the data input y_{n-1}. Adder 914 then adds this result to t_a for phase error detection or zero for frequency detection, and delay circuit 916 delays the result by two time periods so that signal 918 becomes $\mathrm{SGN}\ (y_{n-1})\ *\ y_{n-2} + t_a$ or $\mathrm{SGN}\ (y_{n-1})\ *\ y_{n-2} + 0$.

The circuit surrounded by dashed line 931 produces the other part of the equation:

$-\ \mathrm{SGN}\ (y_{n-1})\ *\ y_{n-1}$

For side sampled acquisition and:

$-\ \mathrm{SGN}\ (y_n)\ *\ y_n$

for center sampled acquisition mode and:

$+\ \mathrm{SGN}\ (y_{n-1})\ *\ y_n$

for frequency detection.

If side sampling read mode is being used, AND circuit 923 and multiplexer circuit 928 select y_{n-1} for input to the EXCLUSIVE OR circuit 930. If center sampling or lock to reference mode is being used, these circuits select y_n for input to the EXCLUSIVE OR circuit 930. EXCLUSIVE OR circuit 927 uses the sign bit, that is, bit five, of y_{n-1} and the READ/LOCK signal 812 to determine whether the output of multiplexer 928 should be inverted and a sign bit of one input to adder 934. If read mode is being used and the sign of y_{n-1} is positive, the output of multiplexer 928 is inverted and a sign bit of one is input to adder 934. If frequency mode is being used and the sign of y_{n-1} is positive, the multiplexer output is not inverted, and a sign bit of zero is input to adder 934. The output of EXCLUSIVE OR circuit 930 and the output of inverter 932 are input to the adder 934 which adds the output of EXCLUSIVE OR circuit 930, and carry bit 933 to the signal 918 to produce the phase error or frequency error value on signal 936. Because of the EXCLUSIVE OR circuit 930 inverts the output of multiplexer 928, and inverter 932 creates a carry bit 933, adder 934 sometimes performs subtraction.

Fig. 10 performs the same function as Fig. 9 to produce a phase or frequency error signal 1036, however, it uses the previous sample as input, and then it combines the phase/frequency error outputs 936 of Fig. 9 and 1036 of Fig. 10.

The previous sample inputs to multiplexer 1012 are y_{n-1} rather than the y_n inputs to multiplexer 912, as are the inputs to multiplexer 1028 and EXCLUSIVE OR circuit 1027. The result is that signal 1036 is the phase/frequency error for the sample previous to the sample processed by Fig. 9. The circuits of Fig. 9 and Fig. 10, in combination, process two samples simultaneously.

After the two phase/frequency errors are determined, the NAND circuits 1038, 1040, and 1042 combine the signals to produce the output 1044 which is sent to Fig. 12 to be combined with the tracking mode signals before being sent to the loop filter 808. The NAND circuits 1038, and 1040 select either the output 936 from Fig. 9 or the output 1036 from Fig. 10 depending upon whether a pulse occurred at time n or at time n-1. These outputs are then connected to NAND circuit 1042 which produces the output 1044.

Figs. 11-12 show the phase/frequency detector circuit for tracking mode. Referring now to Fig. 11, NAND circuit 1106 provides the term "-a * P_{n-3}" for center sampling. For side sampling the output of NAND circuit 1106 is zero. Multiplexer 1114 selects "t_t + c * p_{n-2}" if a pulse occurred at time n-2 or "t_t" alone if a pulse did not occur at time n-2. There are separate set points for acquisition mode and tracking mode. The term "a * P_{n-3}" is subtracted from the output of multiplexer 1114 by using the output of NAND gate 1106, which inverted the value of "a", and inserted a sign bit of one into the adder 1118 through signal 1107 as well as a carry bit of one through signal 1147. Therefore, signal 1120 represents "t_t + c * P_{n-2} - a * P_{n-3}", although the term "-a * P_{n-3}" will only be present for center sampling.

Multiplexer 1124 selects the value y_n for center sampling and y_{n-1} for side sampling and inverts the value before sending it to adder 1122. A carry bit of one is input to the adder 1122 through signal 1121 to cause either y_n or y_{n-1} to be subtracted from y_{n-2}, signal 1020. EXCLUSIVE OR circuit 1126 adjusts for the sign of y_{n-1} and adder 1128 combines the result with signal 1120 to produce the partial equation:

$$SGN\ y_{n-1} * (\ y_{n-2} - y_n) + t_t + c * P_{n-2} - a * P_{n-3}$$

Although the term "-a x P_{n-3}" will only be present for center sampling. Delay circuit 1130 delays this result by two sample times and then adder circuit 1138 subtracts either the constant "a"

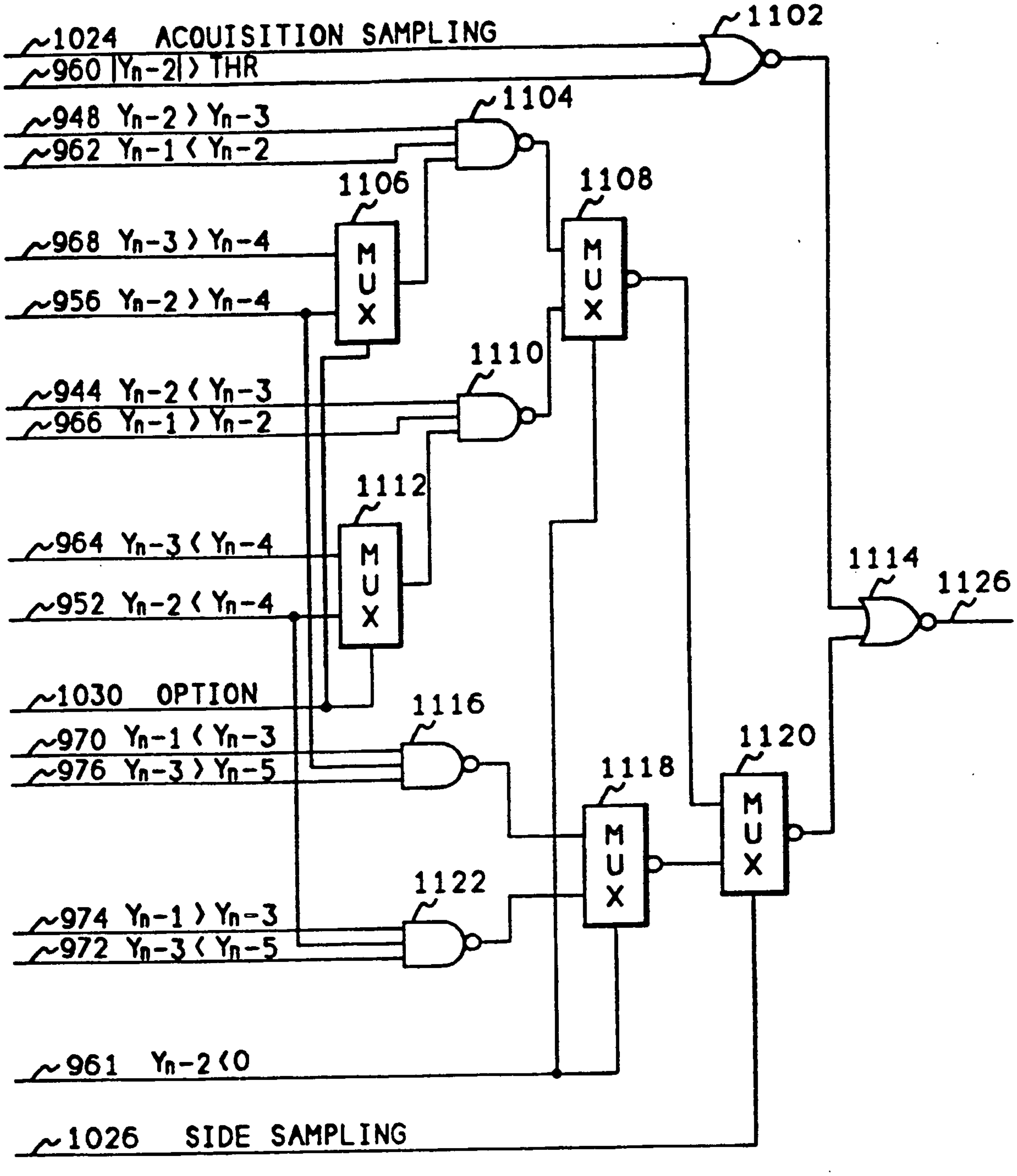

1024 ACQUISITION SAMPLING
960 |Yn-2| > THR
948 Yn-2 > Yn-3
962 Yn-1 < Yn-2
968 Yn-3 > Yn-4
956 Yn-2 > Yn-4
944 Yn-2 < Yn-3
966 Yn-1 > Yn-2
964 Yn-3 < Yn-4
952 Yn-2 < Yn-4
1030 OPTION
970 Yn-1 < Yn-3
976 Yn-3 > Yn-5
974 Yn-1 > Yn-3
972 Yn-3 < Yn-5
961 Yn-2 < 0
1026 SIDE SAMPLING
1102
1104
1106
1108
1110
1112
1114
1116
1118
1120
1122
1126
MUX
MUX
MUX
MUX
MUX
MUX

FIG. 11

BACKGROUND OF THE INVENTION

In a magnetic disk or tape data storage device, data is commonly stored on a magnetic medium by saturation recording in which each portion of the medium is magnetized to the point of saturation in one of two directions. The data to be stored is typically encoded to satisfy certain constraints and the encoded data is used to modulate the direction of magnetization. In a coded representation known as NRZI, each "one" bit of the encoded data causes a transition in the direction of magnetization, while each "zero" bit of the encoded data causes the magnetization direction to remain unchanged. A clock signal is used to write a sequence of encoded NRZI bits as a recording head moves along a track on the medium such that one bit is written at each clock tick.

When a read head is passed over the recorded data track, a voltage pulse is produced at each transition in magnetization. Successive voltage pulses have opposite polarity since successive magnetic transitions are in opposite directions. The written NRZI data sequence may be reconstructed from the resulting voltage waveform by associating a "one" bit with every clock tick at which a pulse occurs and a "zero" bit with every clock tick at which no pulse occurs. The original user data may then be decoded from the NRZI data.

Similar waveforms composed of pulses sometimes occur in other by those skilled in the art that many changes in construction and circuitry and widely differing embodiments and applications of the invention will suggest themselves without departing from the spirit and scope of the present invention. For instance the invention could be implemented with analog circuitry in place of much of the digital circuitry of the preferred embodiment. The disclosures and the description herein are intended to be illustrative and are not in any sense limiting of the invention, more preferably defined in scope by the following claims.

Table 1

	Acquisition	Tracking
Side Sampled	$SGN(y_{n-1})*(y_{n-2}-y_{n-1})+t_a$	$SGN(y_{n-1})*(y_{n-2}-y_{n-1})+t_t+c*P_{n-2}$ $-a*P_{n+2}$
Center Sampled	$SGN(y_{n-1})*(y_{n-2}-y_n)+t_a$	$SGN(y_{n-1})*(y_{n-2}-y_n)+t_t+c*P_{n-2}$ $-b*P_{n+2}-a*P_{n-3}$

Table 2

	ACQUISITION PATTERN	NOMINAL t_a	NOMINAL t_t
Side Sampled	1010...	1 - b + c - a	1 - b
	100100...	1 - b	
	10001000...	1 - b	
Center Sampled	1010...	2 * (c - b)	c - b
	100100...	c - b + a	
	10001000...	c - b	

Table 3

Frequency Detector = $SGN(Y_{n-1})*(Y_n+Y_{n-2})$

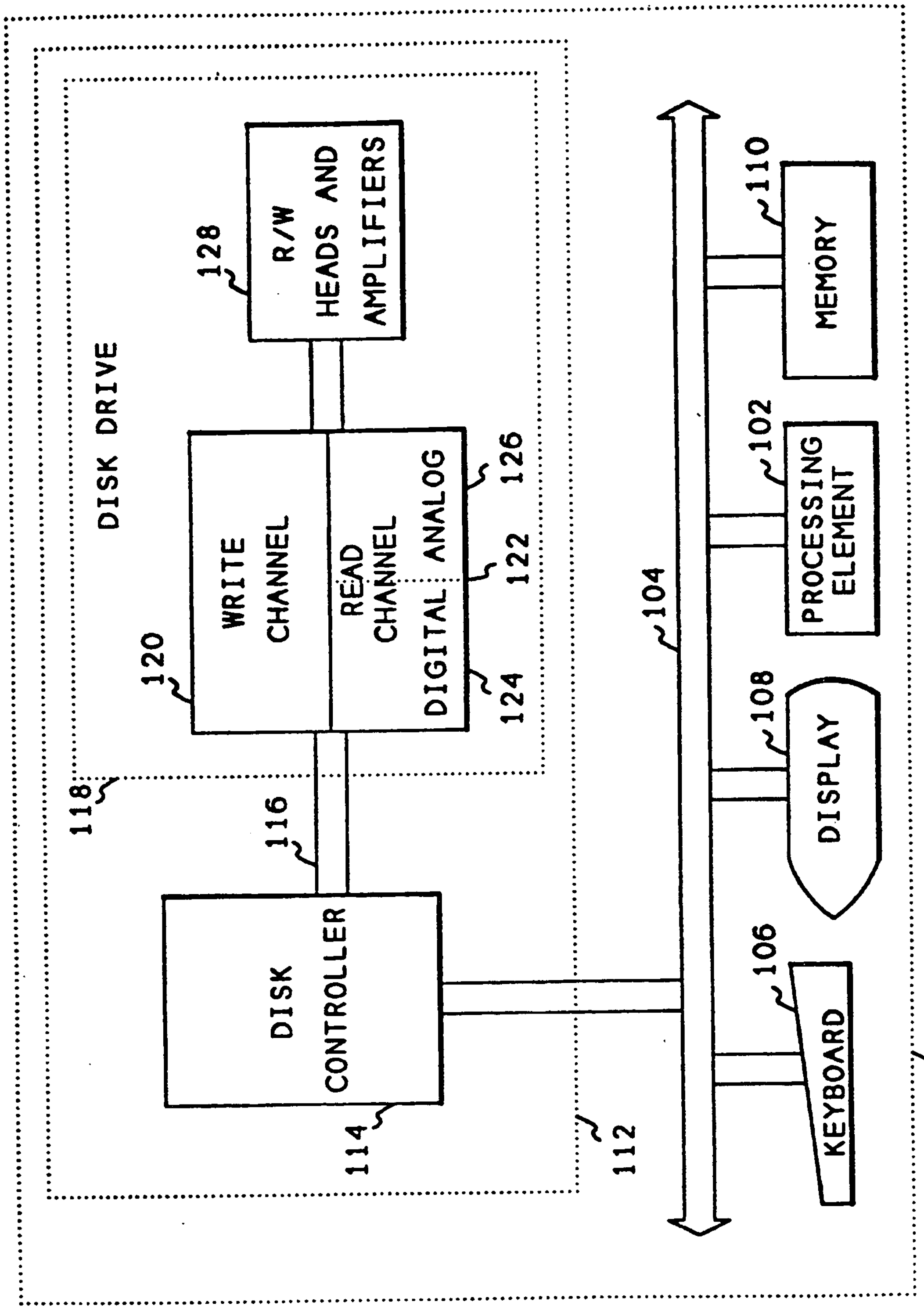

DISK DRIVE
R/W HEADS AND AMPLIFIERS
128
WRITE CHANNEL
READ CHANNEL
DIGITAL
ANALOG
120
122
124
126
118
116
DISK CONTROLLER
114
112
104
MEMORY
110
PROCESSING ELEMENT
102
DISPLAY
108
KEYBOARD
106
100

FIG. 1

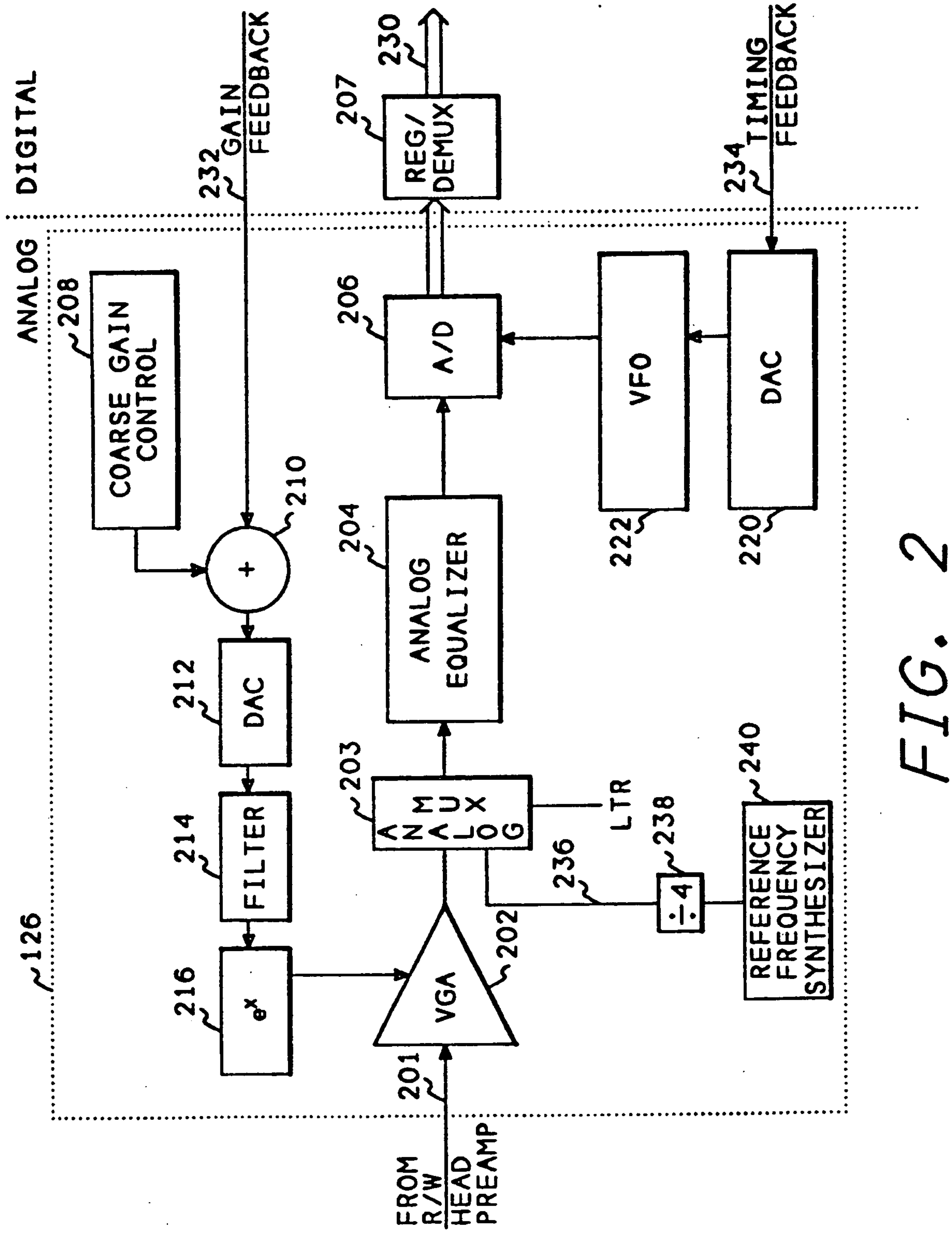

DIGITAL
ANALOG
126
208
COARSE GAIN CONTROL
232
GAIN FEEDBACK
210
+
212
DAC
214
FILTER
216
e^x
VGA
202
201
FROM R/W HEAD PREAMP
203
ANALOG MUX
LTR
236
238
÷4
240
REFERENCE FREQUENCY SYNTHESIZER
204
ANALOG EQUALIZER
206
A/D
207
REG/DEMUX
230
222
VFO
220
DAC
234
TIMING FEEDBACK

FIG. 2

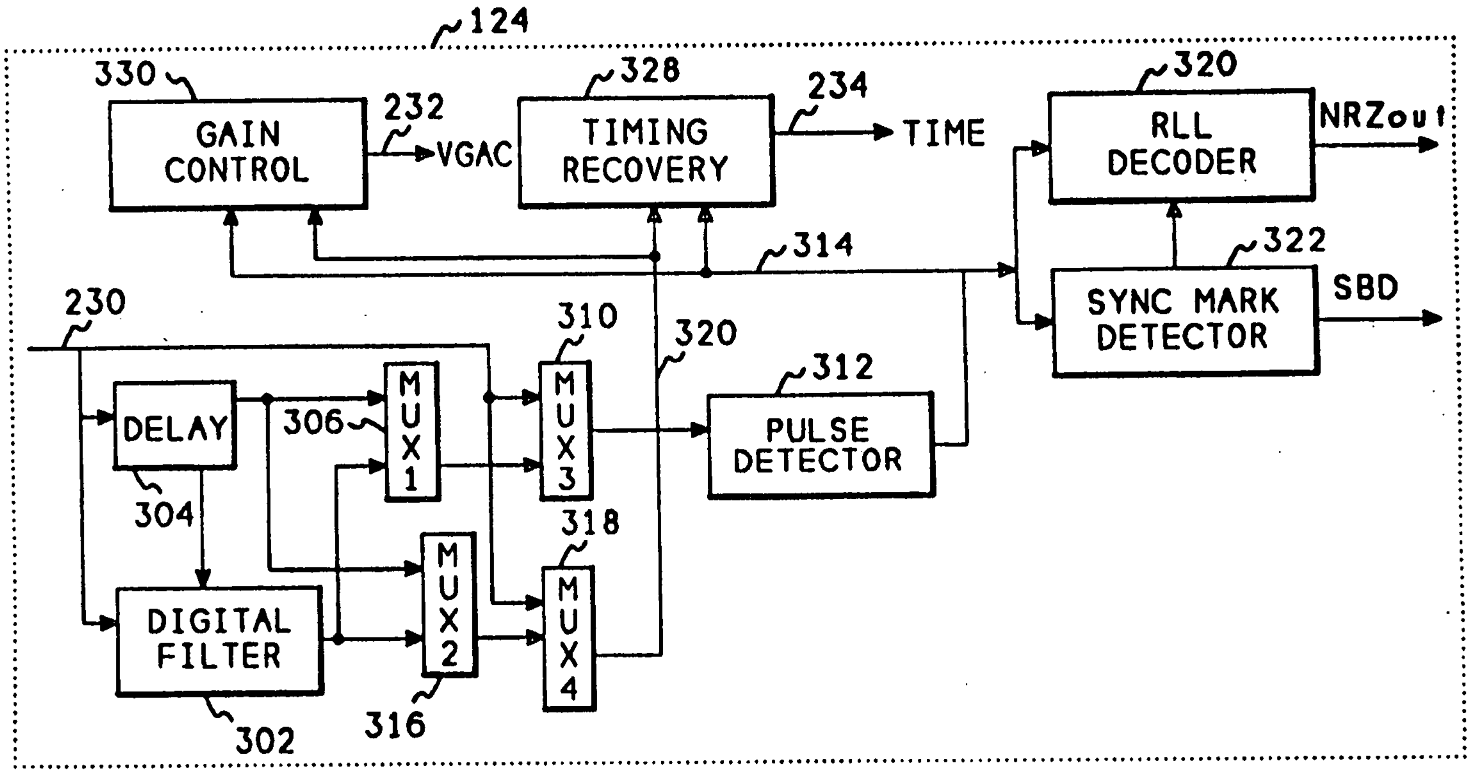

124
330
GAIN CONTROL
232
VGAC
328
TIMING RECOVERY
234
TIME
320
RLL DECODER
NRZout
314
322
SYNC MARK DETECTOR
SBD
230
310
320
DELAY
306
MUX 1
MUX 3
312
PULSE DETECTOR
304
318
DIGITAL FILTER
MUX 2
MUX 4
302
316

FIG. 3

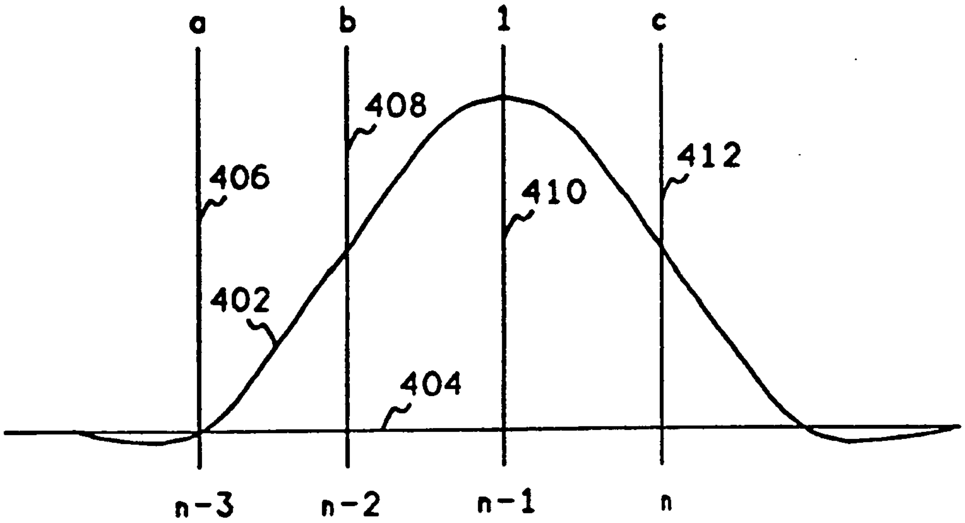

a
b
1
c
408
406
410
412
402
404
n-3
n-2
n-1
n

FIG. 4

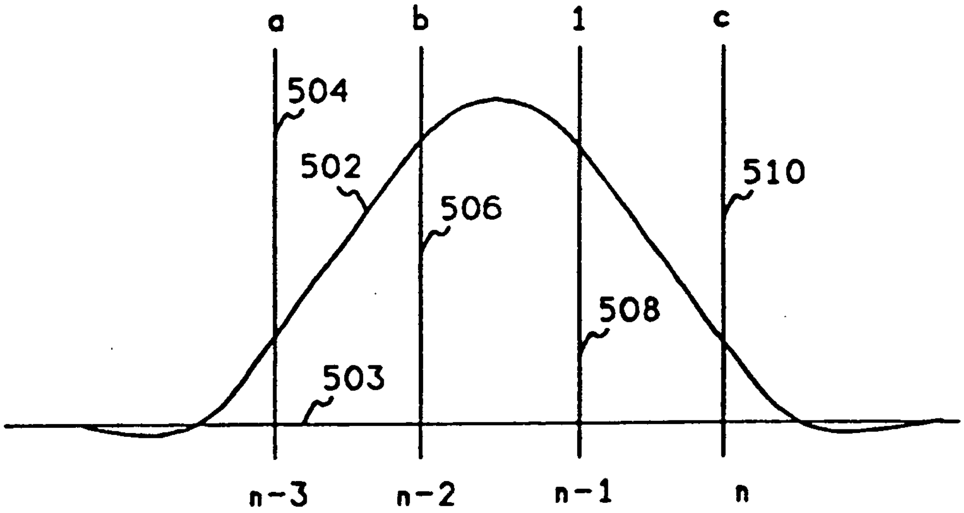

a
b
1
c
504
502
506
510
508
503
n-3
n-2
n-1
n

FIG. 5

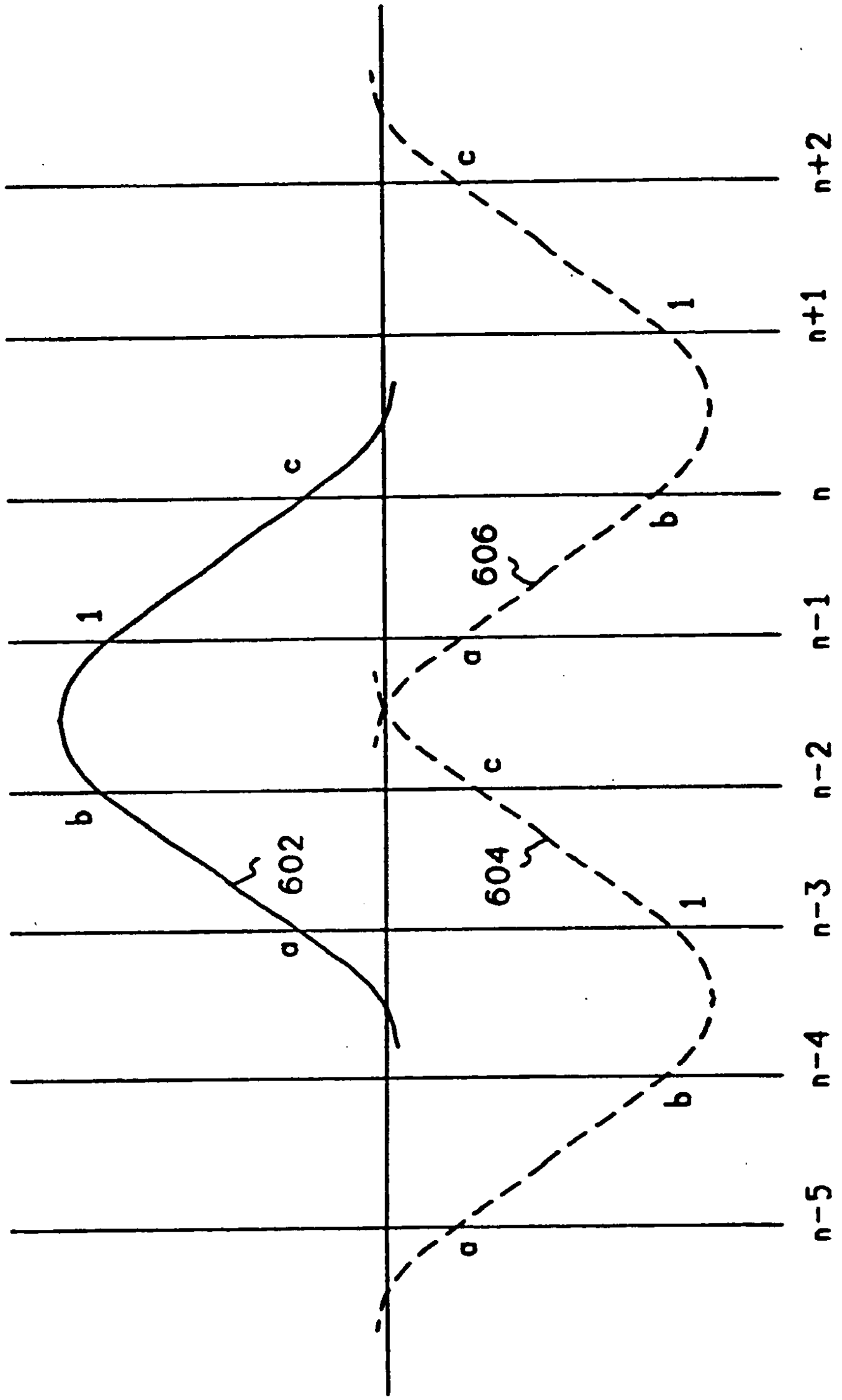


FIG. 6

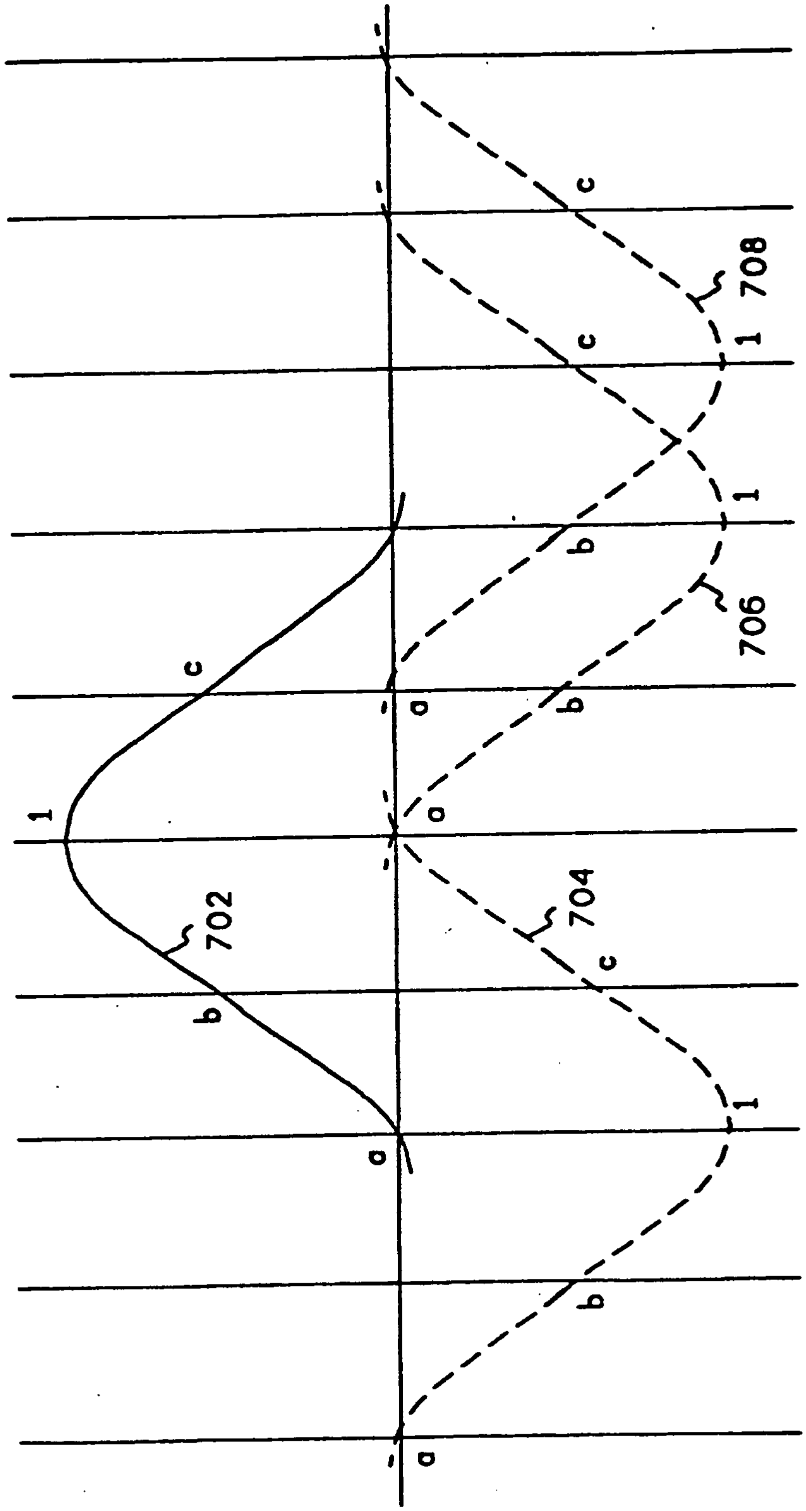


FIG. 7

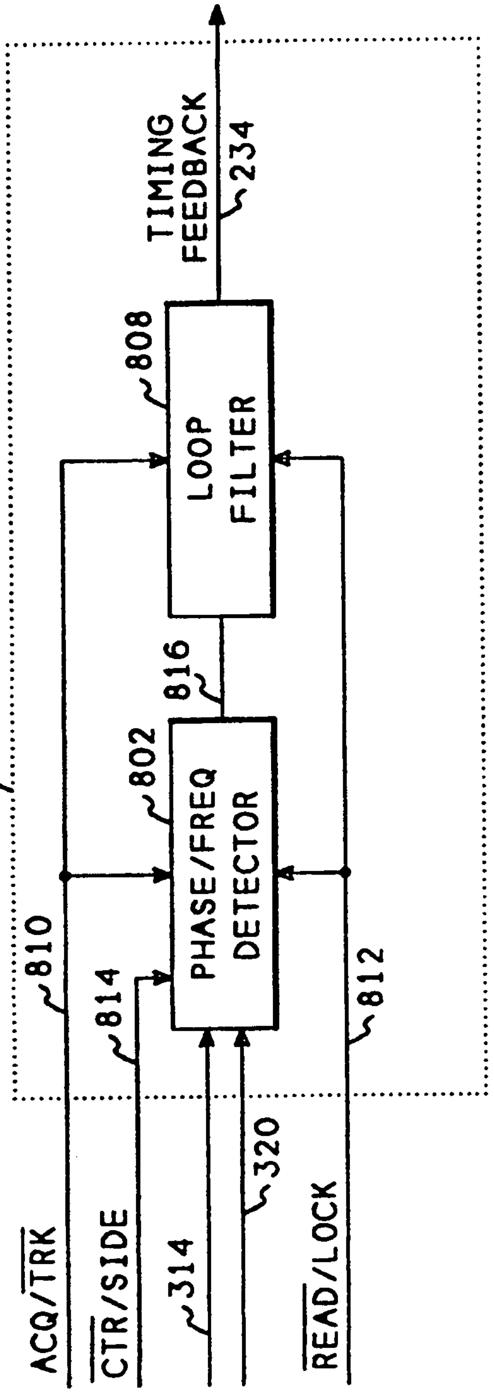

TIMING FEEDBACK
234
808
LOOP FILTER
816
328
802
PHASE/FREQ DETECTOR
810
814
812
ACQ/$\overline{\text{TRK}}$
$\overline{\text{CTR}}$/SIDE
314
320
$\overline{\text{READ}}$/LOCK

FIG. 8

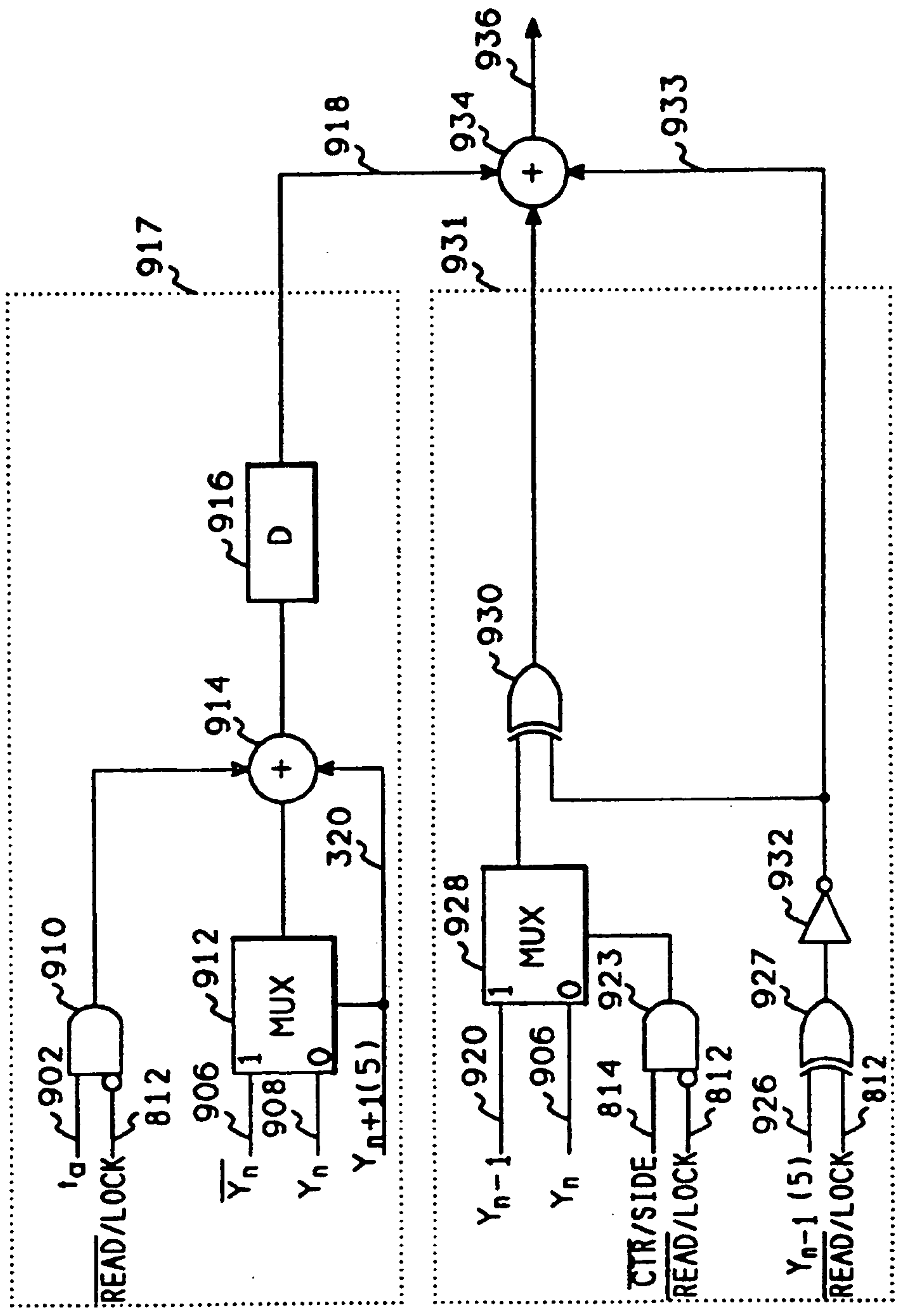

917
916
D
914
+
910
902
912
MUX
1
0
906
908
812
320
918
934
936
933
931
930
928
MUX
1
0
923
932
927
920
906
814
812
926
812
Ia
READ/LOCK
Yn
Yn
Yn+1(5)
Yn-1
Yn
CTR/SIDE
READ/LOCK
Yn-1 (5)
READ/LOCK

FIG. 9

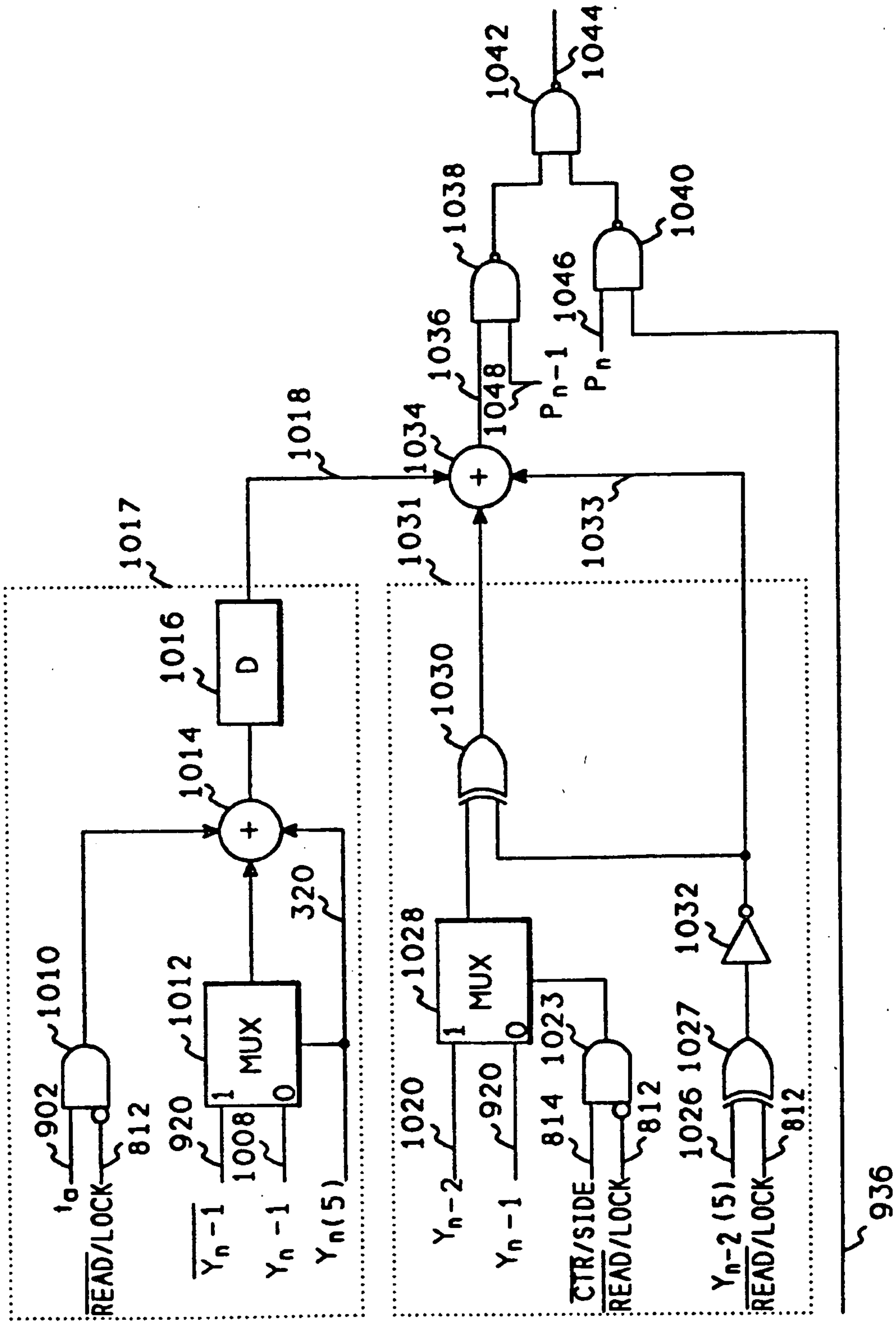
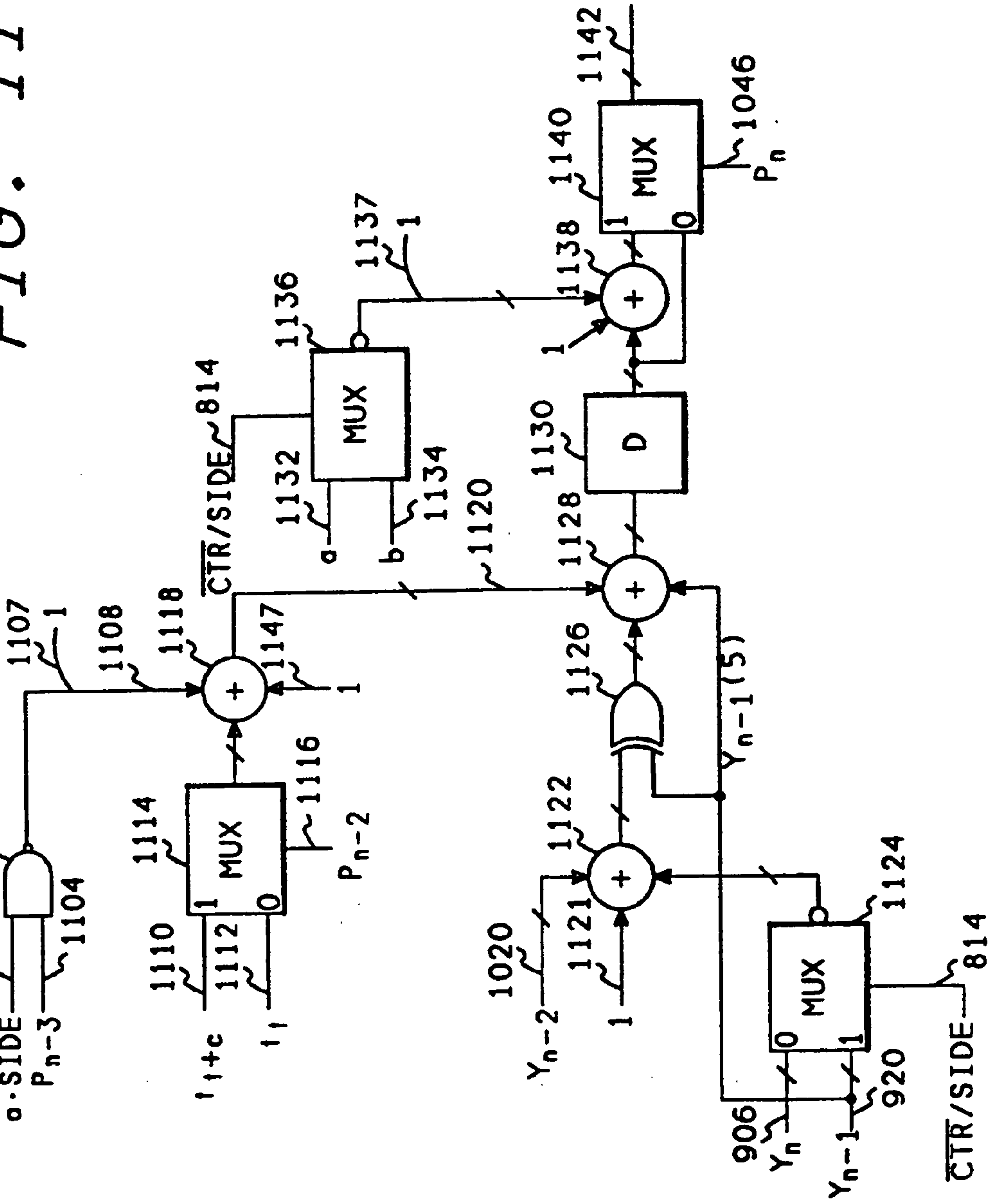

1017
902
1010
t_a
READ/LOCK
812
MUX
1012
920
1008
1
0
$\overline{Y_n-1}$
Y_n-1
$Y_n(5)$
320
1014
+
1016
D
1018
1031
1020
920
814
812
1023
1028
MUX
1
0
Y_n-2
Y_n-1
CTR/SIDE
READ/LOCK
1026
1027
812
$Y_n-2(5)$
READ/LOCK
1032
1030
1033
1034
+
1036
1048
P_n-1
1038
1046
P_n
1040
1042
1044
936

FIG. 10

FIG. 11


a·SIDE
1102
Pn-3
1104
1106
1107
1
1108
1118
1147
1
1110
1112
1+c
1
1114
MUX
1
0
1116
Pn-2
CTR/SIDE
814
1136
MUX
1132
a
1134
b
1120
1137
1
1138
1
1140
MUX
1
0
1046
Pn
1142
1128
1130
D
1126
Yn-1 (5)
1020
Yn-2
1121
1
1122
1124
MUX
0
1
814
CTR/SIDE
906
Yn
920
Yn-1

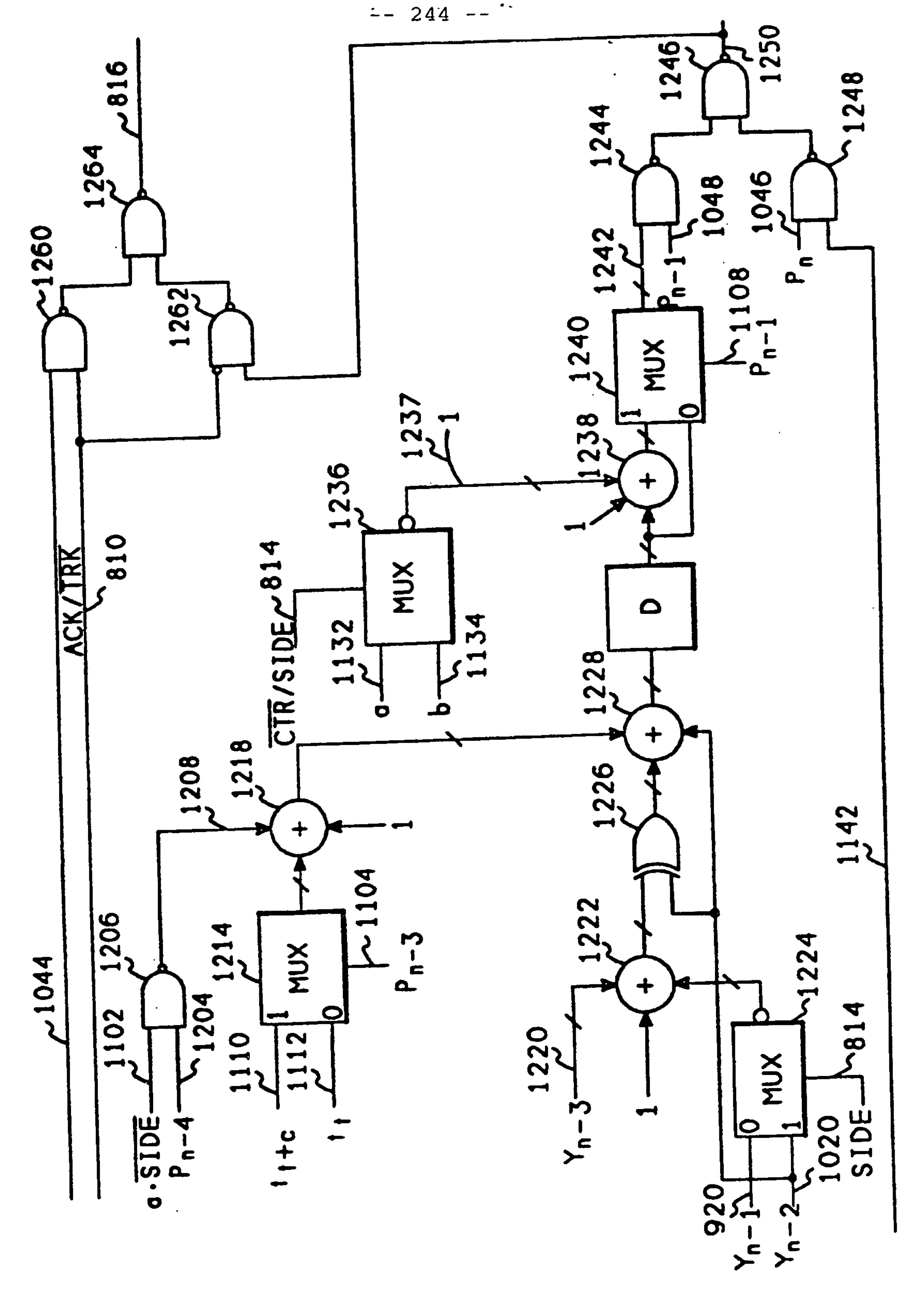

1044
1102
a·SIDE
1206
1204
Pn-4
1110
It+c
1112
It
1214
MUX
1
0
1104
Pn-3
1208
1218
1
ACK/TRK
810
CTR/SIDE
814
1132
a
1134
b
MUX
1236
1237
1
1238
1
1240
MUX
1
0
1108
Pn-1
1242
Pn-1
1048
1244
1246
1250
1248
1046
Pn
1260
1262
1264
816
1220
Yn-3
1222
1
1224
MUX
0
1
814
SIDE
1226
1228
D
1142
920
Yn-1
Yn-2
1020

FIG. 12

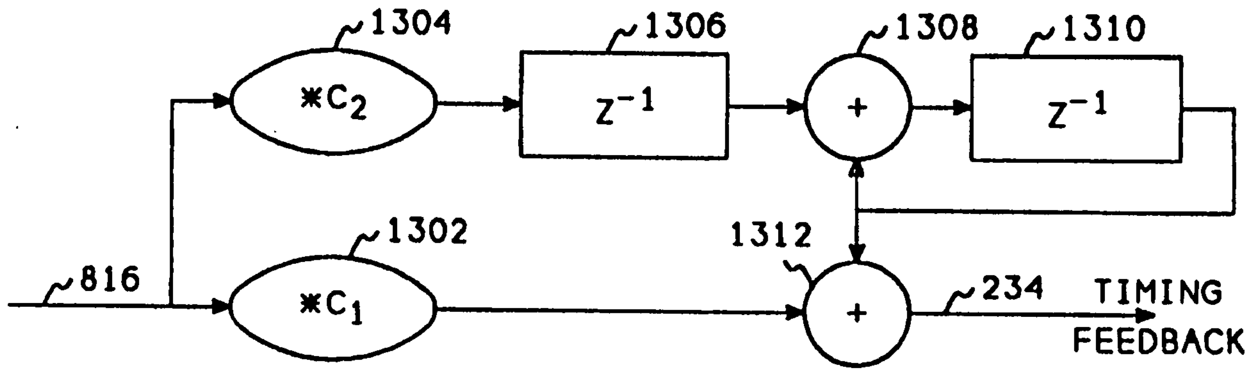

1304
*C$_2$
1306
Z^{-1}
1308
+
1310
Z^{-1}
816
1302
*C$_1$
1312
+
234
TIMING
FEEDBACK

FIG. 13

Our Ref.: 1~~21~~95.P006
42

UNITED STATES PATENT APPLICATION

FOR

METHOD AND APPARATUS FOR REDUCED-COMPLEXITY VITERBI-TYPE SEQUENCE DETECTORS

Inventor(s): Richard T. Behrens
Kent D. Anderson
Neal Glover

Prepared By:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025-1026
(310) 207-3800

*I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Label No*RB620202275 *addressed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231, on* March 16, 1992

Signature *Date*

BACKGROUND OF THE INVENTION

1. Field of the Invention:

This invention relates generally to detection, demodulation and decoding of digital information in data storage and/or communication systems, and to the efficient implementation of such circuits. More particularly, the invention relates to sequence-based demodulation of partial-response signals and to sequence-based decoding of convolutional codes using a Viterbi-like algorithm.

2. Prior Art:

In the storage or transmission of digital information, the bits or symbols of the user data are actually transmitted or stored via a physical medium or mechanism whose responses are essentially analog in nature. The analog write or transmit signal going into the storage/transmission medium or channel is typically modulated by channel bits that are an encoded version of the original user-data bits. The analog read or receive signal coming from the medium is demodulated to detect or extract estimated channel bits, which are then decoded into estimated user-data bits. Ideally, the estimated user-data bits would be an identical copy of the original user-data bits. In practice, they can be corrupted by distortion, timing variations, noise and flaws in the medium and in the write/transmit and read/receive channels.

The process of demodulating the analog read signal into a stream of estimated user-data bits can be implemented digitally. Digital demodulation in advanced mass storage systems requires that the analog read signal be sampled at a rate that is on the order of the channel-bit rate. Maximum-likelihood (ML) demodulation is a process of constructing a best estimate of the channel bits that were written based on digitized samples captured from the analog read signal.

FIGURE 1 shows an exemplary read signal 100, which is a positive-going pulse generated by an inductive read head, for example, from a single medium transition such as transition 103 from North-South to South-North magnetization of track 104 on a rotating disk. Typically, the write signal modulates a transition in the state of the medium to write a channel bit of 1 and modulates the absence of a medium transition to write a 0 channel bit. Thus, transition 103 corresponds to a single channel bit of value 1 in a stream of 0's.

It is common to use run-length-limited (RLL) encoding of the original user data bits, which are arbitrary or unconstrained, into an RLL-encoded stream of channel bits. It may be desirable that there be no less than d zeroes between ones; that is, that the medium transitions be spaced by at least d+1 bit times. This constraint can help keep to a manageable level the interference effects among the pulses in the analog read signal. On the other hand, because medium transitions provide timing information that must be extracted from the read signal to ensure synchronization of the demodulator with the pulses in the read signal, it may be desirable that there be no more than k zeroes between ones; that is, that there be a medium transition at least every k'th bit time. An RLL(d,k) code is a code that can encode an arbitrary stream of original user-data bits into a stream of channel bits such that the encoded channel bit stream satisfies these two constraints.

For example, the following table shows a commonly used RLL(2,7) code:

User Data Bits	RLL(2,7)-Encoded Channel Bits
000	000100
0010	00100100
0011	00001000
010	100100

011	001000
10	0100
11	1000

Note that this RLL(2,7) code requires that the information bit rate available for channel bits be twice the bit rate provided for user-data bits. Other RLL codes enforce the constraint that d=1 (i.e. that no adjacent channel bits both be 1). RLL(1,k) encoding is also called one-constraint encoding, or encoding with a minimum run length of 1.

In FIGURE 1, sample set 101 shows the values of four samples in the case of side sampling of read signal 100; i.e. 0.333, 1.0, 1.0, and 0.333. Sample set 101 is equivalent to the set 1, 3, 3, 1; that is, only the ratios among samples are significant. A signal model gives rise to an expected sample sequence for a single or isolated transition in medium state. Typically, only a few samples of the response to an isolated medium transition are non-zero; in this case, four are non-zero. In a side-sampled signal model such as 1, 3, 3, 1, timing circuitry in the demodulator attempts to maintain a lock on the incoming signal such that there are two adjacent samples on opposite sides of the peak of an isolated pulse. Other sample timing arrangements may be useful. In center sampling, the timing circuitry trys to lock the sample times to the read signal pulses such that one sample occurs at the peak of each pulse. Sample set 102 shows the values of four samples in the case of center sampling of a similar read signal 105; i.e. 0.5, 1.0, 0.5, and 0.0 (or 1.0, 2.0, 1.0 and 0.0 depending on the arbitrary normalization used). An expected sample sequence of 1, 2, 1 corresponds to the signal model known in the prior art as Extended Partial-Response Class IV (EPR4).

Such sample sequences are samples of a continuous-time analog read-signal waveform such as may be produced in the receiver of a communications system or in the readback circuitry of a storage device. For a system that is bandwidth limited to 1/(2T), where T is the sample spacing in time, the sampling theorem declares that the continuous time waveform must be a superposition of sinc functions (sinc(x) is defined as sin(x)/x for x<>0, and as 1 for x=0), with one sinc function centered at each sample point and of amplitude equal to that sample value and with zero crossings at all other sample points. As an example, in saturation magnetic recording, the current in an inductive write head takes on values of +I and -I. The basic excitation applied to the recording channel is a step in current from +I to -I, or vice versa, in the analog write signal. This step in write current produces a transition in the magnetization state of the medium as it moves past the head. When an inductive read head is passed over this magnetic medium transition, a voltage pulse is induced by the bandwidth limited differentiating interaction of the head with the magnetization of the medium. By suitable filtering or equalization, the sequence of samples on an isolated transition response pulse can be made to be {..., 0, 0, 1, 2, 1, 0, 0, ...}, in which case the recording or transmission channel matches the EPR4 signal model.

Another sample sequence well known in the prior art is the Partial Response Class IV signal model (PR4), which corresponds to an expected sample sequence of 0, 1, 1, 0. There are numerous other known signal models. Further, as one is designing or taking measurements on a write/medium/read channel, it may be desirable to take into account the exact response, noise and distortion characteristics of the channel in selecting the signal model to be implemented in the demodulator. Thus, there is a need for a demodulator that is programmable as to the signal model, or expected sequence of sample values for an isolated medium transition.

In data storage it sometimes happens that a data record is not read correctly on the first attempt. Such an event is usually detected by an error detection scheme, but it may not be possible to correct the error(s). In that event the only hope of recovering the data is to attempt another read of the data record. On second and subsequent reads (i.e. on retrys) it may be desirable to change some parameters of the read circuitry in the hopes that such changes will suppress or attenuate the error mechanism. Thus there is need for read-channel electronics that can be easily changed for retries. Additionally, in the case of data storage on rotating disks, the constant angular velocity of the disk results in a variable velocity of the recording medium with respect to the read and write heads. Even if zones are used, in which the channel bit rate varies among zones as the heads move in and out on the disk, there is still a variation in signal shape within each zone from its innermost track to its outermost track. Thus there is a need for read-channel electronics that can be easily changed to accommodate changes in system characteristics.

In situations such as mass information storage in magnetic medium, significant storage-system speed and capacity gains can be realized if the information bits can be closer together in position/time on the medium. However, according to information theory the sample rate must be at least as high as the channel bit rate to utilize all available channel capacity. More precisely, the Nyquist sampling criterion requires that the sample frequency be at least twice the highest frequency contained in the signal, or else information is lost due to aliasing. This information loss could be prevented by introducing redundancy through appropriate coding, but this would reduce the channel capacity for user information. All the partial response signal models described herein may contain frequencies up to one-half of the channel bit rate, implying that the sample rate must be no less than the channel bit rate to avoid aliasing. Sampling at exactly the channel bit rate satisfies the Nyquist criterion when the sample times are appropriately synchronized with the signal. Sampling at the channel bit rate is also convenient for synchronization of the demodulator circuits because the demodulator then produces one estimated channel bit per sample.

Therefore, at least one sample of the analog read signal is typically required per channel bit that is to be demodulated from the signal. Digital decoders are typically complex circuits and may require a slower processing or clock rate than can be supported by analog-to-digital converters and simple buffers. Thus, there is a need for demodulator circuitry that can process in real time read-signal samples that are taken at a higher sampling rate than the processing rate of the demodulator itself.

Further, as medium transitions are more closely positioned, the writing and reading processes become more sensitive to the distortion, timing variations and noise that are inevitably introduced in the processes of writing, storing, and reading. Also, as the transitions become closer, the ability of the medium to fully transition from, say, North-South magnetization to South-North magnetization may be taxed. Also, as the medium transitions become closer, interference effects increase among adjacent or nearby transitions.

FIGURE 2 shows how positive-going pulse 200 from first medium transition 201 combines with negative-going pulse 202 from second transition 203 to produce analog read signal 204, which can be viewed as the interference of the two pulses. Adjacent medium transitions always give rise to read pulses of opposite polarities because they always are created by transitions of opposite types, for example North-South changes to South-North in transition 201, so adjacent transition 202 must be South-North changing back to North-South. Read signal 204 might give rise to a sequence of samples such as 0.333, 1.0, 0.667, -0.667, -1.0, -0.333. To the extent that the read process is linear (and it may not be entirely linear), the voltage waveform induced in the read head will be the superposition of a sequence of pulses, where each pulse is the response to an isolated magnetic transition on the medium.

Clearly, engineering a high-performance digital demodulator is a complex challenge given the combined effects of the limited sampling rate in a digital demodulator, possibly incomplete transitions in the medium, interference among read-signal responses to medium transitions, and distortion, timing variations, noise and flaws in the medium and in the write and read channels.

The prior art uses a method known as partial-response signaling to increase medium transition rates. Partial-response signaling is described in the book "Digital Transmission of Information", by Richard E. Blahut, 1990, pp. 139-158 and 249-255. This method allows the analog response of the storage/transmission medium and of the write/transmit and read/receive circuitry to a medium transition to overlap with the response to adjacent transitions associated with subsequent information bits. If properly implemented, this method can achieve higher information bit rates/densities than the alternative of requiring the medium transitions to be spaced such that the read signal responses do not overlap significantly. A sequence demodulator is required for partial-response signaling.

The prior art uses the Viterbi algorithm to implement sequence detectors, including demodulators and decoders. The Viterbi algorithm is described in the book "Fast Algorithms for Digital Signal Processing", by Richard E. Blahut, 1985, pp. 387-399. A Viterbi demodulator does not attempt to decide whether or not a medium transition has occurred immediately upon receipt of the sample(s) that correspond to that transition. Rather, as samples are taken from the analog read signal, the Viterbi demodulator keeps a running tally of the error between the actual sample sequence and the sample sequence that would be expected if the medium had been written with a particular sequence of transitions. Such an error tally is simultaneously kept for several possible transition sequences. As more samples are taken, less likely choices for transition sequences are pruned from consideration. If the set of possible sequences of medium transitions is appropriately constrained, then the location of each medium transition becomes known with a high degree of likelihood within a reasonable time after taking the samples corresponding to that transition. This effect is shown in FIGURE 5, which illustrates the deferred decision making of a particular Viterbi demodulator in the case of a particular actual sample sequence by showing how the contents of the path memories evolve as additional read-signal samples are taken. A path memory of a sequence demodulator stores information concerning a particular sequence of transitions that is currently being considered as a candidate for the correct transition sequence.

Two of the most significant decisions in designing a modulator and corresponding demodulator are the choice of encoding constraints and the choice of signal models. The encoding constraints chosen may effect the complexity of the demodulator. The filtering and sampling strategy used in the read/receive/demodulate processes can be designed to generate a pulse response to an isolated medium transition that corresponds with the signal model chosen. A sample sequence model is a particular finite-state machine, where the states and transitions of the finite-state machine are determined by the encoding constraints and the signal model chosen. A sequence of expected read-signal samples can be viewed as being generated by a sample sequence model. Viterbi demodulators keep track of one error tally per state in the sample sequence model.

FIGURE 3 shows the sample sequence model for a stream of RLL(1,infinity) encoded channel bits as viewed by a demodulator that uses the EPR4 signal model. Each sequence-model transition is represented in FIGURE 3 by an arrow labeled both with the expected values for the associated read-signal sample and with the current RLL-encoded channel bit that generated the current medium transition, or lack thereof, in the write/transmit/modulate process. For example, sequence-model transition 301 is labeled with expected sample value 302 (+0.5) and with estimated channel bit value 303 (1). Transition 301 occurs upon taking the sample that shows the leading edge of the positive-going read-signal pulse associated with an isolated medium transition, as might be produced for example in the case of a magnetic medium transition from North-South magnetization to South-North magnetization. Each sequence-model state is represented in FIGURE 3 by a circle which, as an aid to understanding, is labeled with a sequence of 3 digits (0 for North and 1 for South) corresponding with the current medium state and the medium states associated with the previous two samples (from right to left). Accordingly, sequence-model transition 301 is the transition from state 000 (or North, North, North) to state 001 (or North, North, South). Note that these state labels do not directly correspond to the sequence of channel bits. Often the sequence-model states are referred to by the decimal equivalents of the state labels used in FIGURE 3 as interpreted as three bit binary numbers. All possible sequences of EPR4 signals that can be read from medium written with an RLL constraint of d=1 may be generated by traversing the state diagram of this sample sequence model.

In general, the number of states in a sample sequence model without RLL constraints is 2^N, where N is the number of samples between the first and last nonzero samples (inclusive) of the system response to its basic excitation. The imposition of coding constraints, such as RLL constraints or other codes mapping from user-data bits to channel bits, may change the number of states and transitions between states. For example, in the case of FIGURE 3, because of the RLL constraint of d=1 chosen, the 010 and 101 states are impossible or prohibited states. To take another case, an RLL constraint of k=7 that is incorporated into the sample sequence model may increase the number of sequence-model states. If each sequence-model transition is assumed to occur with some specified probability, then a sample sequence model is equivalent to a Markov model for a source of expected samples.

The labels on the sequence-model transitions shown in FIGURE 3 associate the presence (estimated channel bit value of 1) or absence (estimated channel bit value of 0) of a medium transition with the sample where the corresponding read-signal response first appears. There are other ways to associate the estimated channel bits to be output with the sequence-model transitions, e.g. the 1 values could appear on sequence-model transitions that correspond to the peak of the read-signal response.

To understand this sample sequence model, consider a simple decoder for an ideal read signal without noise, distortion or timing variations that is implemented according to the state machine of FIGURE 3. This decoder makes one state transition per each sample of the read signal. Assume that this decoder is currently in state 000 (corresponding to a medium sequence of

North-South, North-South, North-South, or NNN) and the demodulator receives a sample value of +0.5. This sample indicates the leading edge of a medium transition from North-South magnetization to South-North magnetization. This results in a next state of 001 (or NNS) and a demodulator output or estimated channel bit of 1, which indicates the presence of this medium transition. Because of the one-constraint encoding, the only sample that can be expected to occur next is +1.0, which is the sample that is aligned with the center of the read-signal pulse due to this medium transition. This sample results in a next state of 011 (or NSS) and a demodulator output of 0, which indicates that another medium-transition response does not start with this sample. Because we have now satisfied the run-length constraint, the next sample may or may not be affected by the leading edge of a pulse due to a second medium transition. Thus state 011 has two possible transitions leading from it. If a second medium-transition pulse is starting with the next sample, then the +0.5 trailing sample that must be present due to the first transition would be offset by the -0.5 leading sample of the second medium transition, and the expected sample would be 0.0. Therefore, receiving a sample value of 0.0 results in a next state of 110 (SSN), and an estimated channel bit of 1 to indicate the second medium transition. Alternatively, receiving a sample value of +0.5 indicates that there is not yet another medium transition and results in a next state of 111 (SSS) and a demodulator output of 0. If the state machine is in state 111, then receiving a sample of 0.0 results in a next state of 111 and an output of 0, i.e. no new medium transition. The remaining transitions and states are symmetrical with those in the above description. Given an ideal set of samples, the output of a demodulator that directly implements this state machine would exactly reproduce the channel bits written to the medium.

In operation, a Viterbi demodulator can be thought of as walking the trellis formed by taking each state of a sample sequence model over each sample time. FIGURE 4 shows the fundamental frame of the trellis of the EPR4 state machine shown in FIGURE 3 between time T and time T+1. Each possible transition of the EPR4 state machine is represented as a branch in the trellis frame. Let S(T) be the read-signal sample taken at time T. Let the branch error metric of, for example, the branch from state 001 to 011, be the quantity (S(T) - 1.0) squared, because +1.0 is the expected sample value for that branch. Because there is only one sequence-model transition leading into state 011, the path error metric for state 011 at time T+1 is always the path error metric for state 001 at time T, plus this branch error metric and the associated estimated channel bit is always 0. To take a second example, state 001 could be reached at time T+1 either from state

000 with a channel bit of 1 and an expected sample of +0.5 or it could be reached from state 100 with a channel bit of 1 and an expected sample of 0.0. So for state 001, both branch error metrics are computed, i.e. the quantity (S(T) - 0.5)^2 and S(T)^2. The first branch error metric is added to the path error metric for state 000 at time T, and the second branch error metric is added to the path error metric for state 100 at time T. The Viterbi algorithm then compares these two metrics and the path with the smallest distance from the actual sample sequence is selected as indicating the most likely path along which state 001 might be entered at time T+1.

For each sample time, the expected sample for each possible state-machine transition, or trellis branch, is compared with the read-signal sample to generate an error metric for that branch. This branch error metric is accumulated over multiple sample times, thus forming a path error metric. A Euclidean distance metric may be used; i.e. a branch error is the square of the difference between the actual read-signal value and the expected value for that state transition at that sample time. A Euclidean path error would be the square root of the sum of the branch errors along that path, but since only comparisons among path error metrics are significant, there is no need to compute a square root and the sum of the branch paths may be used as the path error metric. Other error metrics may be used, for example the absolute value of the difference between the actual sample and the expected sample. For each state at each sample time, the history of possible paths that may have led to that state is reduced by assuming that the state was entered by that path leading into it that has the lowest path metric. This simplification is justified by the fact that no future samples will shed any further light on which path may have led up to that state at that time. As an example of this calculation, state 000 in FIGURE 3 could be entered from either itself or from state 100. In the former case, the expected sample would be 0.0 and in the latter it would be -0.5. At each sample time, the current sample is compared with each of these expected sample values. Let S be the current sample and P(X) the current path error metric associated with state X. If (S + 0.5)^2 + P(100) is less than (S - 0.0)^2 + P(000), then the Viterbi algorithm considers that state 000 would be entered from state 100 at this time and not from the loop from itself. A similar computation is performed for each state at each sample time.

At any sample time, the state that currently has the minimum path error metric could be taken as the correct state and the estimated channel bit could be taken directly from the sequence-model transition corresponding to the chosen branch into that state. But instead, the Viterbi demodulator, like other sequence demodulators, defers this decision until a sequence of subsequent samples has been taken. No attempt is made to determine which state correctly models the channel bit sequence written into the medium at the corresponding write times. Rather, the pruning of unlikely past possibilities occurs only within each state for each sample. Thus, path histories of estimated channel bits must be kept for each state in the sample sequence model. If the modulator and demodulator are designed appropriately for the characteristics of the medium and the read and write channels, then after a reasonable number of samples beyond sample T, the path histories associated with all of the sequence-model states are very likely to make the same estimate of the value of the channel bit corresponding to sample T.

FIGURE 5 illustrates the deferred decisions that a Viterbi demodulator makes. This figure was generated by a software implementation of a particular Viterbi demodulator operating on a particular actual sample sequence. Signal 501 represents a digitally sampled ideal read signal without noise, distortion or timing variations. Trellis path 502 is the correct path, i.e. it corresponds to the written channel bits and to ideal read signal 501. Signal 503 is a noise signal that is added to signal 501 to produce actual sample sequence 504. Trellis 506 shows the paths contained in the path memory, after the 10th sample has been taken, for each of the 8 states of the particular sample sequence model used in this figure. Note that after the 10th sample, each path estimates the same sequence-model state corresponding to the 1st through the 5th samples, but that after the 5th sample, the contents of the path memories diverge in order to find the most likely path into the state that they represent at the 10th sample time. Trellis drawings 507-510 show the paths contained in all path memories as of the 11th through the 14th sample time respectively. In each of trellis drawings 506-510, the path histories associated with all of the sequence-model states make the same estimate of all channel bits up to a reasonably recent time, i.e. between 3 to 5 samples prior to the current sample.

In a mathematical sense, the maximum-likelihood decision rule says to choose as the estimate of the written channel bit sequence that sequence, out of all possible channel bit sequences, for which the conditional probability of receiving the actual sample sequence is highest. A Viterbi demodulator sampling the analog read signal at the channel bit rate satisfies this maximum-likelihood criterion if the analog read signal contains white Gaussian noise added to the expected read-signal pulse samples and the analog read signal is passed through a filter with a frequency response matched to the channel.

For each state in the sample sequence model, the path histories of estimated channel bits are kept in a path memory. The path memory in a sequence detector stores the information necessary to define each surviving path in the trellis. In a prior-art Viterbi detector, there is one surviving path for each state of the source model. The information in the path memory may be encoded and managed in a variety of ways. One way is to simply store the sequence of decisions from each ACS module regarding which of its input paths was chosen at each iteration. In this case the path memory amounts to a set of shift registers, and some means must be provided to trace the paths back through the trellis to determine the estimated channel bit sequence that is the primary output of the demodulator. The length of the path memory is the number of frames back through the trellis for which each surviving path can be reconstructed, counting the current frame. In well-designed detectors, there is a high probability that the surviving paths will all emanate from the same sequence-model state at some point in the trellis within the length of the path memory, and thus share a common path before that point. When this is so, any one of the surviving paths may be traced back to determine the demodulator's estimated channel bit output associated with the oldest frame in the path memory.

In practice, a path memory of 6 to 30 bits may be sufficient. That is, it may be large enough to ensure that there is a high probability that the channel bits are identical in each state's path memory for the bits at the least-recent end of each path memory. Path memory requirements depend on the signal model and on the coding constraints applied. For example, a lower value for k in the RLL(d,k) constraints on the channel bits can in some cases lower the length of the path memory required because the occurrence of medium transitions tends to force or expedite the Viterbi algorithm to make significant decisions.

Path memories can be burdensome to implement. Thus there is a need for techniques to reduce the number of path memories required to implement a Viterbi-like algorithm.

In typical prior-art implementations of the Viterbi algorithm, the signed add, square, add, compare and select computation described above is performed for each sequence-model state. The results of the signed add and square computation may be the branch error metric for more than one branch. In this case, the output of some of the modules that implement the computation of the branch error metrics may be used as inputs to more than one add, compare, select (ACS) module. For example, FIGURE 6 is a block diagram of an implementation of these steps for the EPR4, d=1 sample sequence model, which is shown in FIGURE 3. Branch error metric generators 600-604 implement the signed add and square computation for each of the possible values of expected samples, i..e. -1.0, -0.5, 0, +0.5, and +1.0). ACS modules 605, 606, 608, and 609 implement the add, compare, and select computation for states 000, 001, 111, and 110 respectively. Each also stores the current path error metric for the best path, at any given time, into the state it represents. They also communicate, via selection indicator lines 611-614 respectively, to path memories 615 which of the two possible input transitions they have selected as the most likely path into the state they represent. A more detailed block diagram of the implementation of two-input ACS modules 605, 606, 608, and 609 is shown in FIGURE 7. Modules 607 and 610 are ACS modules for the special case of states 011 and 100 which have only one sequence-model transition leading into them. Modules 607 and 610 implement the add and store-path-error-metric functions only. In this prior-art Viterbi demodulator implementation, one ACS module is required for each state in the sample sequence model. ACS modules can be burdensome to implement.

As more samples are taken per each medium transition response and as larger amounts of interference are allowed among pulses in the read signal, more states are required in the sample sequence model. Thus there is a need for a technique that reduces the for side sampling, or the constant "b" for center sampling, before sending the result to multiplexer 1140. Multiplexer 1140 selects either the output of delay circuit 1130 or the output of adder circuit 1138 depending upon whether a peak occurred at time n. Since delay circuit 1130 has delayed by two clock cycles, signal P_n 1046 actually represents P_{n-2} and therefore determines whether the constant terms a or b should be added in. Therefore, output 1142 is the solution to the tracking equations of Table 1 for the current sample.

Fig. 12 is identical to the circuit of Fig. 11 except that the samples used are delayed by one, and additional circuitry exists to combine the two phase errors. For example, the inputs to multiplexer 1224 and adder 1222 are one sample behind the inputs to the equivalent circuits in Fig. 11. Therefore, the output 1242 ~~of~~ provides the solution to the tracking equations of Table 1 delayed by one sample time.

R.B. 9-
TDD 9-
JG 9-

Fig. 12 then combines the phase error results 1142 and 1242 of Figs. 11 and 12, respectively, to provide the phase/frequency detector output 816 to the loop filter. The NAND circuits 1244 and 1248 select the output 1142 of Fig. 11 or the output 1242 of Fig. 12 depending upon whether a pulse occurred at time n-2 or time n-3. This result is sent to NAND circuit 1246 which produces the output 1250 which is combined with the acquisition mode signals before being connected to the loop filter.

NAND circuits 1260, 1262, and 1264 select the acquisition phase/frequency error signal 1052 or the tracking phase error signal 1250, depending upon the TRK/ACQ signal to create the phase/frequency circuit output 816 which is sent to the loop filter.

Fig. 13 shows a diagram of the loop filter 808. The z-domain transfer function of this filter is

$$F_t(z) = (c_1 * z * (z - 1) + c_2) / (z * (z - 1)$$

where z^{-1} represents a delay of two channel bit intervals (each $1/f_s$) because this digital filter operates at one-half the channel sample rate. Coefficients c_1 1302 and c_2 1304 are independently programmable for acquisition and tracking. For lock to reference mode, c_1 is set to zero and c_2 is independently programmable. A means is also provided to change the significance of the loop filter output between acquisition and tracking modes since the appropriate range and resolution of frequency control depends on which mode is used. Separate registers are provided for each mode to facilitate this change. Delay element 1306 is used for pipelining and has little effect on the filter response.

A linear discrete-time approximation to the open-loop transfer function of the loop is

$$G_t(z) = (K_p * K_o * (c_1 * z^2 - c_1 * z + c_2) / (z^{n+1} * (z - 1)^2)$$

where K_p is the gain of the phase detector (which depends upon the pulse shape), K_o is the control gain of the VCO and n is the number of $(2/f_s)$ clock delays in the loop.

Having thus described a presently preferred embodiment of the present invention, it will now be appreciated that the aspects of the invention have been fully achieved, and it will be understood

010000 with its squared path metric of 0.2402, over the channel-bit sequence of 001000 with its squared path metric of 0.2602.

If one of the paths in FIGURE 8 is the correct path, i.e. the path representing the sequence-model transitions that correspond to the sequence of written channel bits, then the other path is an error path. The selection by the demodulator of the error path instead of the correct path would constitute an actual error event, and would result in one or more erroneous estimated channel bits being output from the demodulator. The example of FIGURE 8 is a closed potential error event beginning at time T+1 and ending at time T+5 because the erroneous path and the correct path converge to the same state at time T+5. In contrast, it would be an open potential error event if the trellis ended at time T+4, before the paths had merged.

More generally, any path through the trellis may be the correct path and any pair of paths in the trellis constitutes a potential error event as long as they begin in the same state. A potential error event is any two paths in the trellis that begin in the same state and diverge in their second state. The beginning of a potential error event is the point where any two paths diverge. A codeword is the expected sample sequence associated with a particular trellis path.

The distance of a potential error event is the distance between the codewords of the two paths of the potential error event. Using the Euclidean metric, the square of this distance is equal to the path error metric that would be built up in the error path if the actual sample sequence corresponded to the ideal sequence for the correct path. The minimum distance for a given signal model is defined as the minimum distance of any closed potential error event or the minimum distance of any open potential error event of duration greater than the path length, whichever is smaller. Since the error rate of the demodulator decreases with increased minimum distance, one is motivated to choose, if possible, a path length sufficient to contain all open error events whose distance is less than the minimum distance of closed potential error events. These considerations are analogous to Hamming distance properties of error correction codes.

At relatively low bit error rates, the performance of a Viterbi demodulator is almost entirely determined by the minimum distance and by the signal-to-noise ratio. Actual error events of distance larger than the minimum distance are quite improbable compared with the minimum distance events and may safely be ignored in the design of the demodulator without seriously affecting the bit error rate. Thus there is a need for a technique that reduces the implementation complexity of the Viterbi algorithm without sacrificing its ability to handle minimum distance error events.

Partial-Response Polynomials

A partial-response information channel is ideally equivalent to a finite impulse response (FIR) digital filter. For example, if the input to a magnetic disk recording channel is taken as a sequence of write current levels taking values of plus and minus I, then the unit pulse response of the digital filter is the response to the write current sequence

-I -I -I -I +I -I -I -I -I

divided by 2I (the amplitude of the current pulse). The DC offset in the above unit pulse is immaterial if the channel response is 0 at DC. This is the case with magnetic disk recording channels using inductive read heads. In other cases, the DC offset can be easily corrected for by subtracting the response to a DC current of -I from the response to the above current pulse. For an EPR4 channel the response to the current pulse above would be as follows:

...0 0 1 2 1 0 0 0 ... (response to first transition)

plus ...0 0 0 -1 -2 -1 0 0 ... (response to second transition)

to give...0 0 1 1 -1 -1 0 0 ... (response to pulse)

Since an ideal partial-response system is equivalent to a discrete-time filter, the analysis techniques used on discrete-time filters can be applied to partial-response systems. One such analysis technique is the Z-transform, in which the inverse of the transform variable z represents a time delay by one sample interval. A similar analysis technique commonly used in magnetic disk storage is the D-transform wherein the transform variable D represents the same delay as the inverse of z. The Z-transform (or D-transform) of the unit pulse response of a digital filter is the transfer function of that filter, and may be used to reveal its characteristics in the frequency domain. For a finite impulse response filter, the D-transform is simply the polynomial in D whose coefficients are the same as the unit pulse response sequence. For the EPR4 unit pulse response sequence of (1, 1, -1, -1), the D-transform is the polynomial 1 + D - D^2 - D^3. This polynomial is referred to as the EPR4 partial-response polynomial. This polynomial factors to (1-D)(1+D)^2, which reveals that the EPR4 channel has a spectral null at DC and a second order spectral null at the Nyquist frequency of one half the sample rate.

The response of a partial-response channel to a unit pulse can be computed by taking the inverse D-transform of the partial-response polynomial. This amounts to reading the sequence of polynomial coefficients and taking them as the sequence of sample values. It is also possible to determine the response to an isolated transition, otherwise known as the step response of the system, from the partial-response polynomial. This is done by dividing the polynomial by (1-D) and taking the inverse D-transform. Dividing out the (1-D) factor from the EPR4 polynomial leaves the polynomial (1 + 2D + D^2), which corresponds to the EPR4 isolated pulse of 1, 2, 1.

Convolutional Codes

In addition to their use for demodulating partial-response signals, Viterbi detectors can also be used to decode convolutional codes. Viterbi decoding of convolutional codes is described in the book "Error-Correction Coding for Digital Communications", by George C. Clark Jr., and J. Bibb Cain, 1981, pp. 228-235. The encoding process for a convolutional code may be described as a convolution operation between the data symbol stream and an encoding sequence, called a generator. This encoding process may also be viewed as a filtering operation where the generator is the finite impulse response of a discrete time filter. The convolution or filtering is frequently done with finite field arithmetic (often binary). Redundancy may be introduced into convolutional codes in order to allow error detection and/or correction capabilities by using multiple generators. The encoding filters for the generators operate in parallel and their outputs may be interleaved into a single encoded data stream. By combining multiple generators with decimation or subsampling of the output sequences from each encoding filter one can introduce redundancy in any desired amount. For example, one could use three generators, tripling the number of information symbols, but discard every second symbol from each generator so that the encoder produces 3 encoded symbols for every 2 data symbols and the code rate is 2/3. Such a rate 2/3 encoder is illustrated in FIGURE 9.

A partial-response data channel is based on the fact that an isolated step or pulse at the input to the channel produces a particular response of finite duration in the sampled output sequence, and that the response to a sequence of input pulses is the linear superposition of the responses to each individual input pulse. This is equivalent to a rate 1 convolutional code in which the generator is the impulse response of the data channel.

Thus there is a need for a technique for reducing the implementation complexity of Viterbi detectors for decoding of convolutional codes as well as those for demodulation of partial-response signals.

BRIEF SUMMARY OF THE INVENTION

A Viterbi detector, such as may be used to demodulate partial-response signals or to decode convolutional codes, is modified to reduce its implementation complexity. In the case of demodulation, a partial-response signal may be viewed as a sequence of expected samples generated from a finite-state-machine model. In a typical Viterbi demodulator implemented using the add, compare, select method, each state in the expected sample sequence model is associated with a hardware module to perform the functions of adding new branch error metrics to path error metrics, comparing path error metrics, and selecting the path having the lowest path error metric. In the prior art, the required number of these add, compare, select (ACS) modules is equal to the number of sequence-model states. In this invention, an ACS module may have two or more sequence-model states dynamically associated with it, such that at some times one sequence-model state is associated with it and at other times another sequence-model state is associated with it. The present invention reduces the number of ACS modules required and also reduces the size/complexity of the demodulator's path memories which must store one path for each ACS module. Groups of sequence-model states may be chosen to share an ACS module without significant loss in performance as compared to the original, unreduced Viterbi demodulator. Further, the present invention supports a wide range of sample models by making the expected sample sequence of an isolated medium transition programmable. Further, the present invention reduces the speed at which the detector circuitry must operate relative to the sample rate by allowing multiple samples to be processed simultaneously. Further, several reduced detectors for specific sample sequence models are presented for particular applications. The present invention is equally applicable to other types of Viterbi detectors, such as the case of decoders for convolutional codes.

It is an object of the invention to provide sequence detectors, such as demodulators for partial-response signals and decoders for convolutional codes, having reduced implementation complexity. Implementation complexity may be taken to mean the amount of processing required, whether that processing is embodied in logic gates, analog circuits, or software.

Another object of the invention is to accomplish the complexity reduction with minimal degradation of performance as compared to a prior-art Viterbi detector, particularly with respect to minimum distance errors and other errors of relatively high likelihood or importance.

A still further object is to provide sequence detectors, demodulators, and decoders that are well suited for implementation in digital integrated circuit form.

Yet another object is to provide sequence detectors, demodulators, and decoders suitable for signals whose source models contain more states than would otherwise be allowable within given limits for detector complexity.

Another object of the invention is to reduce the path memory required to implement a Viterbi-like algorithm.

Another object is to apply these simplifications regardless of the convolutional code, the sample sequence model, the channel bit encoding constraints, or the signal model used.

A still further object is to apply these simplifications to Viterbi-like demodulators for nonlinear channels, as well as to Viterbi demodulators for linear channels.

Another object is to implement a Viterbi-like detector, demodulator, or decoder with a programmable expected sample sequence for an isolated medium transition. A related object is to support multiple, alternative expected sample sequences for retrys or for situations in which the medium characteristics change, e.g. within or among zones of a disk storage device in which a constant angular velocity of the disk results in variable velocity of the medium with respect to the read or write heads. Another related object is to develop programmable sample sequence models that support multiple signal models that would require distinct sample sequence models and distinct hardware architectures when implemented with non-programmable sample sequences.

Another object is to apply these simplifications regardless of the RLL or other user-data-bit to channel-bit encoding used.

Yet another object is to implement a Viterbi-like detector, demodulator, or decoder that can process two or more actual sample values per detector cycle time.

Another object is to implement sequence detectors for several specific sample sequence models and with several specific selected subsets of potential error events by applying the general techniques of this invention.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 shows a state transition on a medium such as a track of a disk drive and its associated pulse in an analog read signal. It also shows two digitized sample models of such read-signal pulses.

FIGURE 2 shows two adjacent medium transitions and their individual and combined read-signal pulses.

FIGURE 3 is the state machine model of the expected sequence of samples for the case of an EPR4 signal model and a d=1 run-length constraint.

FIGURE 4 shows the fundamental frame of the EPR4, d=1 trellis.

FIGURE 5 illustrates the deferred decision making of a particular Viterbi demodulator in the case of a particular actual sample sequence by showing how the contents of the path memories evolve as additional read-signal samples are taken.

FIGURE 6 is a block diagram of a prior-art implementation of a Viterbi demodulator for the EPR4, d=1 sample sequence model.

FIGURE 7 is a more detailed block diagram of the implementation of an add, compare, select (ACS) module for a sequence-model state having two transitions that may enter the state.

FIGURE 8 shows an error event on the EPR4, d=1 trellis.

FIGURE 9 shows a block diagram of a rate 2/3 encoder for a convolutional code suitable for decoding using the Viterbi algorithm.

FIGURE 10 is a sharing diagram where a line between states indicates that that pair of states from the EPR4, d=1 sample sequence model of FIGURE 3 can share an ACS module in the demodulator based on the criterion that all error events of distance less than or equal to 2 must be handled properly, i.e. in the same fashion as would a Viterbi demodulator with one ACS module for each of the 6 states of the sample sequence model.

FIGURE 11 is a version of the sample sequence model from FIGURE 3, showing how a reduced-complexity detector for that signal may be constructed with only two ACS modules that each represent one of three states at any given time.

FIGURE 12 is a block diagram of a simplified implementation of the demodulator shown in FIGURE 6 using only two ACS modules selected according to the sharing diagram of FIGURE 10.

FIGURE 13 is a sample sequence model for a partial-response class IV (PR4) signal without RLL constraints. The corresponding partial-response polynomial is (1 - D) (1 + D), and the expected sample sequence for an isolated transition is 1, 1.

FIGURE 14 is a sharing diagram for the sample sequence model of FIGURE 13 under the condition that all error events of distance less than or equal to the square root of 2 are resolved properly, with a path length of P = 10.

FIGURE 15 is a sharing diagram for the sample sequence model of FIGURE 3 under the conditions that all error events of distance less than or equal to 2 and all error events that correspond to drop-outs or drop-ins are resolved properly with a path length of P = 6.

FIGURE 16 is the sample sequence model for an extended partial-response class IV (EPR4) signal without RLL constraints. The corresponding partial-response polynomial is (1 - D) (1 + D)^2, and the expected sample sequence for an isolated transition is 1, 2, 1.

FIGURE 17 is a sharing diagram for the sample sequence model of FIGURE 16 under the condition that all error events of distance less than or equal to 2 are resolved properly, with a path length of P = 10.

FIGURE 18 is a sample sequence model for a partial-response class IV (PR4) signal with an RLL constraint of d = 1. The expected sample sequence for an isolated transition is 1,1.

FIGURE 19 is a sharing diagram for the sample sequence model of FIGURE 18 under the condition that all error events of distance less than or equal to the square root of 2 are resolved properly, with a path length of P = 10. This same sharing diagram applies when the added condition of proper resolution of drop-in and drop-out error events is applied.

FIGURE 20 is a sample sequence model state diagram for an RLL constraint of d = 1 and a signal model given by the polynomial (1 - D) (1 + D)^3. The corresponding expected sample sequence for an isolated transition is 1, 3, 3, 1.

FIGURE 21 is a sharing diagram for the sample sequence model of FIGURE 20 under the condition that all error events of distance less than or equal to the square root of 10 are resolved properly, with a path length of P = 11.

FIGURE 22 is a block diagram of an implementation of a Viterbi demodulator for the sample sequence model of FIGURE 20 simplified to use only three ACS modules as taken from the sharing diagram of FIGURE 21.

FIGURE 23 is a block diagram of an implementation of a Viterbi demodulator for the sample sequence model of FIGURE 20 simplified to use five ACS modules as taken from the sharing diagram of FIGURE 21.

FIGURE 24 is a sharing diagram like FIGURE 21 with the added condition that all drop-in and drop-out error events are resolved properly.

FIGURE 25 is a generic block diagram of a sequence demodulator with shared ACS modules.

FIGURE 26 is a sample sequence model for a nonlinear write/medium/read system.

FIGURE 27 is a sharing diagram for the sample sequence model of FIGURE 26.

Figure 28 illustrates a six state machine model with a programmable expected sequence of samples.

Figure 29 illustrates a ten state machine model with another programmable expected sequence of samples.

FIGURE 30 shows a six-state modified sample sequence model for EPR4 with a minimum run length constraint of d=1, each sequence model transition in Figure 30 representing the combination of two consecutive sequence model transitions in the unmodified sample sequence model of FIGURE 3.

FIGURE 31 is a sharing diagram for Figure 30 for the condition that all potential error events of distance less than or equal to 2 be handled properly.

FIGURE 32 is a diagram showing how states from FIGURE 30 may be grouped to share two ACS modules.

FIGURE 33 shows a ten-state modified sample sequence model for partial response signals corresponding to the polynomial (1-D)(1+D)^3 with a minimum run length constraint of d=1, each sequence model transition in Figure 33 representing the combination of two consecutive sequence model transitions in the original model of FIGURE 20.

FIGURE 34 is a diagram showing how states from FIGURE 33 may be grouped to share three ACS modules.

FIGURE 35 is a diagram showing how states from FIGURE 33 may be grouped to share five ACS modules.

FIGURE 36 shows a modified four-state sample sequence model for PR4 signals without run length constraints, differing from FIGURE 13 in that each sequence model transition in this model represents the combination of two consecutive sequence model transitions in the original model of FIGURE 13.

FIGURE 37 is a diagram showing how states from FIGURE 36 may be grouped to share two ACS modules.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

The present invention can be thought of as exploiting the fact that when, as in the prior art, a separate ACS module is associated with each sequence-model state, then much of the time the ACS modules are computing path error metrics for paths that are so far away from the actual sequence of samples that their computations are very likely to be irrelevant. The present invention dynamically associates a particular ACS module (and its associated path memory) with more than one sequence-model state. One key to practicing the present invention is to determine which states can share an ACS module without significant performance degradation in terms of distinguishing the correct trellis path from those potential error events that are reasonably likely to occur, or that are of special interest for whatever reason.

The following is a step-by-step procedure for determining which states in a sample sequence model may share an ACS module in a sequence demodulator without unacceptable performance degradation. Following the formal description of each step is an example of how that step is applied to the EPR4 d=1 sample sequence model of FIGURE 3. Several of the steps are clearly well suited to being carried out with the aid of a computer. Appendix A lists computer software in MATLAB which may be used to help carry out steps 3 through 5. MATLAB is a computing tool distributed by The MathWorks, Inc. of Natick, Mass.

Step 1. Generate a finite-state-machine model for the expected sample sequence from the read signal to be demodulated. Number the states in some convenient fashion.

FIGURE 3 shows the result of step 1 for the EPR4 model with the d=1 constraint. For convenience we shall refer to the states by the decimal equivalent of their binary labels. For example, an isolated pulse of positive polarity is produced by beginning in state 0, moving to state 1, then 3, then 7, and remaining in state 7 via the self-loop. As this model traverses this sequence of states, the expected sample labels on the paths traversed generate the EPR4 sampled pulse, {0.5, 1.0, 0.5, 0, ...}, and the estimated channel bit labels indicate a single 1 that excited the channel to produce this pulse.

Step 2. Choose a length, P, for the path memory in the demodulator. This path length may be only for analysis purposes, and the path length implemented may be different if desired.

In general one should choose a path length no less than the maximum duration of any open or closed potential error event at minimum distance. Other factors may also be involved in the choice of path length, particularly in cases where there exist open potential error events of infinite length at minimum distance. We choose path length P=6 for our example, which is known to be adequate for the EPR4, d=1 sample sequence model.

Step 3. Make a complete list of all codewords, or legal sequences of expected samples, that are of length P or less. Include in the list the sequence of states that generates each codeword. A codeword of length n is any sequence of expected samples that can be generated by n consecutive state transitions of the state machine from step 1. For this purpose, two codewords are considered distinct if their state sequences differ, even if the expected sample sequence is the same. In general, codewords of length less than the path memory length, P, must be considered in addition to codewords of length P.

It is not practical to include the exhaustive list of EPR4 d=1 codewords of length 6 or less in this document. As an example we list all sample sequences of length 4 that begin in state 0 (000). (In performing this step, all sample sequences of length 4 beginning in any state must be considered, only those starting in state 0 being used herein for illustration purposes.) Note that the state sequences for length 4 sample sequences have 5 entries because they contain both the beginning and ending states.

CODEWORD	STATE SEQUENCE
0 0 0 0	0 0 0 0 0
0 0 0 1	0 0 0 0 1
0 0 1 2	0 0 0 1 3
0 1 2 1	0 0 1 3 7
0 1 2 0	0 0 1 3 6
1 2 1 0	0 1 3 7 7
1 2 1 -1	0 1 3 7 6
1 2 0 -2	0 1 3 6 4

Step 4. Consider all possible potential error events, both open and closed, from the list of sample sequences generated in Step 3. A potential error event is any pair of sample sequences of equal length which have the same starting state but differ in the second state in their sequences. It is possible for the first sample sequence to be the correct (transmitted or stored) sample sequence while the demodulator erroneously chooses the second sample sequence. An error event is "closed" if the two state sequences ever return to the same state at the same time (even if they do not remain together), otherwise it is "open". Choose a

number of ACS modules required to implement a Viterbi-like algorithm.

Consider the following actual sample sequence: 0.0, 0.25, 0.75, 0.75, 0.25, 0.0 in the d=1 EPR4 partial-response model described above. The trellis diagram shown in FIGURE 8 shows two paths that could equally well, or equally poorly, have generated such a sequence of samples (there are other such paths: the two paths shown assume that state 000 was the initial state). The channel bit sequences associated with these two paths are 010000 and 001000. They differ by one bit time in terms of where the medium transition occurs. The expected sample sequence associated with the first of these two paths is 0.0, 0.5, 1.0, 0.5, 0.0, 0.0, while the expected sample sequence for the second path is 0.0, 0.0, 0.5, 1.0, 0.5, 0.0. The squared Euclidean metric for each path may be computed for a given actual sample sequence by summing the squares of the differences between the actual sample sequence and each expected sample sequence. The result is that each path arrives in state 111 at time T+5 with a squared path metric of 0.25. This represents a case where the actual sample sequence lies on a decision boundary, and some arbitrary choice must be made between the two paths. More commonly, the sample values will favor one path over the others, for example an actual sample sequence of 0.0, 0.26, 0.75, 0.75, and 0.24 favors the channel-bit sequence of

1 2 1 -1	0 1 3 7 6	2 (closed)

Step 5. Build a sharing table with one row and one column for each state in the state machine from step 1. Begin with each element of the sharing table TRUE. An element is marked FALSE if the two states represented by its row and its column are ever simultaneously occupied by the two state sequences of any error event in the chosen set. This criterion applies at any position in each error-event state sequence for the subset of potential error events selected in step 4. In the sharing table, either the order of the states can be taken both ways, or alternatively only the upper triangle of the sharing table can be used. All remaining TRUE elements in the sharing table, if any, indicate that the corresponding pair of states may share an ACS module in the demodulator without sacrificing the ability to handle every error event in the chosen subset just as it would be handled in the full Viterbi demodulator.

For the EPR4 d=1 example, we will build a sharing table with 6 rows and 6 columns. Each entry is presumed TRUE until shown to be FALSE. Consider the first error event from the example list in step 4. The state sequence for the first sample sequence is in state 0 at the second clock cycle, while the state sequence for the second sample sequence is in state 1. This means that to distinguish between these two paths and properly process this error event we must not share an ACS module between states 0 and 1. Thus mark the element in the sharing table in the row for state 0 and the column for state 1 as FALSE. Also mark the element in the sharing table in the column for state 0 and the row for state 1 as FALSE. The remainder of the state sequences for the first example error event imply that the pairs (1,3) and (3,7) are also unable to share an ACS module. When this line of reasoning is continued through all error events in the subset selected in step 4, the following sharing table results (where 1 means TRUE and 0 means FALSE):

	(0)	(1)	(3)	(7)	(6)	(4)
	000	001	011	111	110	100
(0) 000	1	0	1	1	1	0
(1) 001	0	1	0	1	1	1
(3) 011	1	0	1	0	1	1
(7) 111	1	1	0	1	0	1
(6) 110	1	1	1	0	1	0
(4) 100	0	1	1	1	0	1

The above sharing table shows whether pairs of states can share an ACS module based on both the error events listed as examples in step 3 and on the additional error events that must be considered but were not listed in the example. When in a full execution of step 3 one considers the sample sequences that start from state 7 (111) and the related error events of step 4, one finds that the two pairs of states (0,4) and (4,6) cannot share an ACS module. It happens that sample sequences that start from states other than 0 or 7 do not yield any additional pairs of states that can not share.

FIGURE 10 shows the information from the above EPR4 d=1 sharing table in a graphical way. Each circle represents one of the 6 states. Each pair of states that are allowed to share an ACS module is connected with a line. A group of states may mutually share an ACS module only if every pair within the group is allowed to share.

Step 6. Decide how to group the states that are allowed to share ACS modules, if any. The states may be grouped to obtain the minimum number of ACS modules possible, or non-minimum groupings may be chosen because of other desirable properties.

The EPR4 d=1 sharing table from step 5 indicates that there is only one grouping possible that uses only two ACS modules. In this grouping, one ACS module is shared by states 000, 011, and 110, while the second ACS module is shared by states 001, 111, and 100.

Other groupings are possible using more than 2 ACS modules. One of particular interest is to share in the pairs: 000 with 111, 001 with 110, and 100 with 011. With this sharing arrangement, the path labels have a fixed magnitude and only depend on the state of origin for their signs, potentially resulting in further simplification of the implementation of the branch-error-metric computation.

To prevent substantial performance loss one should include all minimum distance potential error events in the chosen subset. Other potential error events may be included as desired. For example, one could choose to include all the potential error events whose distance was less than a specified threshold which is greater than the minimum distance. Or one could choose to include all potential error events that can be characterized as resulting from a particular non-ideal aspect of the information channel, such as dropouts from defects in magnetic medium. Potential error events of length less than the path memory length, P, must be considered in addition to error events of length P. This can be seen by first considering error events of length 2. The 2 terminal states of a length 2 error event cannot share an ACS module unless the associated error event at length 2 has already exceeded the threshold distance. It may be possible that all length P extensions of this length 2 error event exceed the threshold distance, but the fact that all extensions exceed the threshold distance is not sufficient for the 2 states to share an ACS module. This argument can be extended to pairs of states on error events of any length less than P. Thus, if such a potential error event were the only one to forbid sharing between a particular pair of states, and if one considered only potential error events of length P, one would erroneously conclude that the pair of states in question were able to share an ACS module. In reality, such sharing would force a premature decision between two paths whose distance was still below the threshold at the time the decision was made. This would reduce the effective minimum distance and increase the error rate accordingly.

This completes the step by step process for determining which states will share ACS modules in a sequence detector. This process may be applied to any state-machine signal model to design a reduced complexity Viterbi-like detector for that signal. By suitable choice of the selected subset of error events, one can control the performance degradation associated with the complexity reduction. It is frequently possible to obtain substantial complexity reduction with negligible performance degradation.

Principles and examples follow which indicate how such detectors may be implemented.

The sharing of ACS modules may be viewed as a dynamic association of the source-model states with the ACS modules. The sharing of ACS modules is implemented by adding one or more memory bits to each shared module to indicate which state the module represents at any given time. The number of memory bits must be sufficient to allow a unique representation for each state sharing the module, thus requiring the number of bits to be greater than or equal to the base-2 logarithm of the number of states sharing the module. At each processing cycle, the branch error metrics for all input paths to a given ACS module are evaluated and added to the corresponding path error metrics stored in other ACS modules. The new path error metrics are compared and the lowest one is selected as the surviving path for that ACS module. The new path error metric for the surviving path is stored in the ACS module for use in the next clock cycle. In addition, the most recent state from which the surviving path comes uniquely defines which state the current ACS module should represent at this time, so its memory bits are set appropriately. The labels on the paths leaving the ACS module, which will be used to generate branch error metrics during the next clock cycle, are determined by the memory bits within the ACS module.

FIGURE 11 is a sample sequence model in which the sequence-model states from FIGURE 3 have been grouped into two groups that share ACS modules based on the sharing diagram of FIGURE 10. ACS module 1100 is associated with one of states 000, 011, or 110 at any given point in time. ACS module 1101 is associated with one of the remaining three states at any given point in time. Transition 1102 either represents the transition from state 000 to itself (if ACS module 1101 currently is in state 000), or it represents the transition from state 011 to state 100 (if ACS module 1101 currently is in state 011). Transition 1102 can be described by the following table:

Current State	Next State	Expected Sample / Estimated Channel Bit
000	000	0.0 / 0
011	110	0.0 / 1

For example, if ACS module 1100 internal state memory bits indicate that it represents state 011 at a particular time, then the self-loop on that module has an expected sample of 0.0, an estimated channel bit of 1. Should that path be selected, then the next state to be represented by that ACS module is 110. Note that if the present state is 110, then the self-loop from ACS module

1100 to itself does not exist and some means must be provided to prevent its selection.

Similarly, transition 1103 can be represented by the following table:

Current State	Next State	Expected Sample / Estimated Channel Bit
000	001	0.5 / 1
011	111	0.5 / 0
110	100	-1.0 / 0

Transition 1104 can be represented by the following table:

Current State	Next State	Expected Sample / Estimated Channel Bit
111	111	0.0 / 0
100	001	0.0 / 1

Note that if the present state is 001, then the self loop does not exist and some means must be provided to prevent its selection.

Transition 1105 can be represented by the following table:

Current State	Next State	Expected Sample / Estimated Channel Bit

001	011	-1.0 / 0
111	110	-0.5 / 1
100	000	-0.5 / 0

Each path in FIGURE 11 is now in effect labeled with a table showing the expected sample labels and the estimated channel bit (to be output to the path memory) labels and the next state for each of the possible states that may be represented by the ACS module from which the path originates.

The block diagram in FIGURE 12 shows an implementation of a reduced-complexity Viterbi demodulator for EPR4 signals with d = 1 simplified in accordance with the sharing diagram of FIGURE 10 and the grouped sample sequence model of FIGURE 11. There are only two ACS modules, each having only two sequence-model transitions to select between and each representing one of three states at any particular time. There are only 4 branch error metric generators even though there are 5 different expected-sample values as path labels. This is possible because the total number of paths in the state machine of FIGURE 11 is only 4, implying that it is never the case that branch error metrics for all 5 expected sample values will be needed simultaneously. The expected sample memory stores the expected sample labels from the tables for each path in FIGURE 11.

FIGURE 13 is a sample sequence model for a partial-response class IV (PR4) signal without RLL constraints. The corresponding partial-response polynomial is (1 - D) (1 + D), and the expected sample sequence for an isolated medium transition is 1, 1. FIGURE 14 is a sharing diagram for the sample sequence model of FIGURE 13 under the condition that all error events of distance less than or equal to the square root of 2 are resolved properly for a path length of P = 10, i.e. are resolved as they would be by a full Viterbi demodulator with one ACS module per sequence-model state. FIGURE 14 shows that there is a unique grouping for a grouped sample sequence model with a minimum number of ACS modules: i.e. states 1 with 2; and states 3 with 4.

FIGURE 15 is a sharing diagram for the sample sequence model of FIGURE 3 (EPR4, d=1) under the conditions that all error events of distance less than or equal to 2 and all error events that correspond to drop-out or drop-in error events (more thoroughly described later) are resolved properly with a path length of P = 6. FIGURE 15 shows that there is a unique minimum-ACS grouping for this case: i.e. states 100 with 001; states 110 with 011; state 000; and state 111.

FIGURE 16 is a sample sequence model for an extended partial-response class IV (EPR4) signal without RLL constraints. The corresponding partial-response polynomial is (1 - D) (1 + D)^2, and the expected sample sequence for an isolated medium transition is 1, 2, 1. FIGURE 17 is a sharing diagram for the signal model of FIGURE 16 under the condition that all error events of distance less than or equal to 2 are resolved properly with a path length of P = 10. There is only one ACS module that can be shared among two states in this case: i.e. states 3 and 4.

FIGURE 18 is a sample sequence model for a partial-response class IV (PR4) signal with an RLL constraint of d = 1. The expected sample sequence for an isolated medium transition is 1, 1. FIGURE 19 is a sharing diagram for the sample sequence model of FIGURE 18 under the condition that all error events of distance less than or equal to the square root of 2 are resolved properly with a path length of P = 10. This same sharing diagram applies when the added condition of proper resolution of drop-in and drop-out error events is applied. Like FIGURE 14, FIGURE 19 shows that there is a unique minimum-ACS grouping: i.e. states 1 with 2; and states 3 with 4.

FIGURE 20 is a sample sequence model for an RLL constraint of d = 1 and a signal model having a partial-response polynomial of (1 - D) (1 + D)^3. The corresponding expected sample sequence for an isolated medium transition is 1, 3, 3, 1.

FIGURE 21 is a sharing diagram for the sample sequence model of FIGURE 20 under the condition that all error events of distance less than or equal to the square root of 10 are resolved as they would be by the full Viterbi demodulator, with a path length of P = 11. The sharing diagram of FIGURE 21 indicates that the states can be grouped in the following three groups each sharing an ACS module: 1, 3, 8, and 10; 2, 4, and 6; 5, 7, and 9. It is also clear that no grouping can be made into less than three sharing groups, since states 1, 2, and 5 have no connections with one another and must end up in separate groups. FIGURE 22 is a block diagram of such a three-ACS implementation.

There are many sharing arrangements for the sample sequence model of FIGURE 20 that have four or more ACS modules. One of special significance is the pairing: 1 and 10, 2 and 9, 3 and 8, 4 and 7, 5 and 6. This arrangement requires five ACS modules to implement, but has the advantage that the magnitude of all expected samples within the same sequence-model transition is constant regardless of the current sequence-model state within the sequence-model state group. This may allow the branch metrics to be partially precomputed with minimal additional hardware, which may reduce the time required for each iteration to be processed, because only the sign of the expected sample depends on the current sequence-model state. FIGURE 23 is a block diagram of such a five-ACS implementation.

FIGURE 24 is a sharing diagram of the sample sequence model of FIGURE 20. It uses the conditions of FIGURE 21 with the added condition that all drop-in and drop-out error events are properly resolved. One minimum-ACS groupings for this sample sequence model requires the following 6 ACS modules: states 1 and 8; states 2 and 6; states 3 and 10; states 5 and 9; state 4; and state 7.

FIGURE 25 is a generic block diagram of a sequence demodulator in which some or all of the ACS modules are associated with multiple states. The state transitions block determines, for each ACS module at each iteration, which of its associated states it should represent at the next iteration based on the selected branch into that ACS module and the present state of the ACS module from which the selected branch originates. The branch label selection block determines the appropriate branch labels for each branch based upon the present state represented by the ACS module from which the branch originates.

In general, trade-offs may need to be made between the complexity reductions allowed by this invention and other considerations such as circuit timing. For example, the 5-ACS implementation of FIGURE 23 may have advantages over the apparently less complex 3-ACS implementation of FIGURE 22.

Drop-out and drop-in error events

One of the non-ideal aspects of some medium, of some magnetic recording medium for example, is the presence of small regions on the medium that do not make transitions, e.g. magnetize, as desired. Such a medium defect can result in the attenuation or complete suppression of the analog read signal over a small time interval. Or, it may result in spurious pulses in the analog read signal caused by the transition between a magnetized region and a defect region. A drop-out error is an error in which a written transition is not detected, while a drop-in error is an error in which a transition is falsely detected where none was written. In a situation where medium defects tend to cause drop-out or drop-in errors, it may be important for the sequence demodulator to handle drop outs and drop ins to the best of its ability. If so, drop-out and drop-in errors need to be characterized and included in the set of error events for which the reduced-complexity detector performs as well as the full Viterbi detector.

We characterize an error event as a drop-out if the estimated channel bit sequence contains a 0 indicating no transition in every position where the correct channel bit sequence contains a 0, and if the estimated channel bit sequence contains zeros in one or more positions where the correct channel bit sequence contains a 1 indicating that a transition was written. An error event is characterized as a drop-in when the same error event would be a drop-out if the roles of the correct and estimated channel bit sequences were reversed. Since either path in an error event is a legal choice for the correct path, the inclusion of all drop-out error events in the subset of interest automatically includes all drop-in error events. The events that are not included are those that represent some combination of drop-outs and drop-ins within the same error event. The sharing diagrams in FIGURE 15 and FIGURE 24 show the allowable sharing of ACS modules for two different signal models when drop-out and drop-in error events are included in addition to minimum distance error events.

Finite State Models for Nonlinear Channels

Partial-response channels are linear systems, but the use of a finite state machine model for a signal source does not necessarily presuppose a linear system. Whether or not a state machine represents a linear system, a Viterbi detector may be built to match the state machine model and select surviving paths through the associated trellis. The present invention applies to such detectors for nonlinear channels just as well as it applies to Viterbi detectors for linear channels.

FIGURE 26 shows a sample sequence model for a system with a nonlinear write/transmit channel, medium response, and/or read/receive channel. The type of nonlinearity described here is a simplification of a nonlinearity actually exhibited by some hard disk medium at high recording densities. The sampled response of this system to an isolated transition is 0, 1, 1, 0, as would be the case for a PR4 channel. But because of system nonlinearities, adjacent transitions mutually reduce the amplitude of the readback pulses for each transition by an amount greater than would be predicted by linear superposition of the pulses. If a transition has a neighboring transition in the adjacent bit positions on either but not both sides, then its response pulse is reduced in amplitude so that the sample values are 0, a, a, 0 for some a<1. If a transition has neighboring transitions in the adjacent bit positions on both sides its amplitude is further reduced to give sample values of 0, b, b, 0 for some b<a. The binary state labels may represent a four-bit record of the state of the medium. If the location of a transition is associated with the first nonzero sample of the resulting pulse, this state machine incorporates one bit look-ahead in the sense that the rightmost digit in the state label indicates the state of the channel in the bit position following the one whose transition response is just beginning. This is necessary in order to determine whether or not the current transition is reduced in amplitude by a succeeding transition. However, this look-ahead is only conceptual, since we could just as well have associated each transition with the sample just before its response pulse began.

The nonlinear behavior modeled in FIGURE 26 reduces the minimum distance for this model as compared to the minimum distance for the PR4 model, which is the square root of 2. It also creates a situation in which there are potential error events of distance slightly greater than minimum distance but close enough to minimum distance that it may be important to include them in the set considered to be of interest in Step 3 of the simplification method above. Therefore, the optimal ACS sharing possibilities in the sample sequence model of FIGURE 26 depend on the nonlinearity parameters a and b, as well as on the distance chosen as the threshold for error events of interest. They may also depend on the coding constraints, such as RLL encoding, and the path-memory length selected. FIGURE 27 shows a sharing diagram for the sample sequence model of FIGURE 26 for the case of a=0.85, b=0.7, P=8, and including all potential error events of distance less than or equal to the square root of 2 (the minimum distance of closed potential error events in this case is 0.85 times the square root of 2). The sharing diagram of FIGURE 27 shows that a minimum-ACS detector for this sample sequence model can be implemented by 10 ACS modules as follows: states 0 with 13; states 1 with 12; states 2 with 15; states 3 with 14, states 6 with 9; states 7 with 8; state 4; state 5; state 10; and state 11.

Simplified Software Implementation of Viterbi Sequence Detectors

In situations where the information rate is low relative to the instruction execution rate of an available processor, it is possible to demodulate samples of an analog read signal using a multipurpose processor executing appropriate software. While such an implementation does not have ACS modules, where a "module" is a hardware unit, is does have ACS routines that perform, as a sequence of instructions on the processor, a series of add, compare, select functions that are equivalent to the functions of a hardware ACS module. The techniques presented herein are applicable to reduce the number of executions of such ACS routines required to process each sample taken, thus speeding up the maximum sample rate that a given processor can support. It is possible that this speed up could, in a particular situation, make a software implementation of a Viterbi demodulator practical whereas without the simplification of the present invention and its associated speedup a hardware demodulator would have been required. Besides the potentially lower implementation cost of such a software demodulator, a software implementation would provide potentially very advantageous flexibility in the choice of sample models and encoding constraints.

Sample-Model Programmability

For some applications, it is preferable to implement a sequence detector such that one can vary the sample sequence expected due to an isolated transition. Figure 28 illustrates a six state machine model with a programmable expected sequence of samples a, b, c, and Figure 29 illustrates a ten state machine model with another programmable expected sequence of samples a, b, 1, c. The sample sequence a, b, 1, c can fit any signal model of 4 or fewer samples per isolated medium transition. The values of a, b, and c should be allowed to range from, say 0 to 2, so that they can be chosen such that the constraint that the third expected sample is 1 is a mere matter of normalization. For example, allowing a and c to range from 0 to approximately 0.5 and allowing b to range from 0 to approximately 2, allows the sequence a, b, 1, c to cover any pulse shape that may reasonably be expected to be produced by inductive magnetic read heads over a range of recording densities.

A significant implication of the programmability of the 4-sample detector involves combining some of the special cases that were treated in the prior art as quite distinct. If one sets a, b, 1, c to 0, 1, 1, 0 one has a detector for PR4. If one sets a, b, 1, c to 0, 0.5, 1, 0.5 one has a detector for EPR4. The point is that PR4 and EPR4 and 4-samples would ordinarily each have their own distinct state machine model, and thus their own architecture of ACS modules. The sample model programmability of the present invention supports the PR4 or the EPR4 (or one of numerous other signal models) cases with a single detector or demodulator architecture.

The use of a programmable sequence detector allows a great variety of retry strategies to be implemented. When used in combination with a programmable equalizer, different equalization targets can be set for both the detector and the equalizer for attempted rereads of a data record. For example, one could begin with the detector set for a side-sampled isolated pulse of 1, 3, 3, 1 and the equalizer programmed to provide such a pulse. Upon failure to read a data record one could then switch to the center-sampled pulse of 1, 2, 1, alter the equalizer to provide such a pulse and attempt to read the data record again.

The use of a programmable sequence detector also has the benefit that the optimum equalization target may not be the same for every part of the recording medium. The inside track of a magnetic disk may need a different pulse shape for best performance than the outside track. Also, for disks with multiple heads and recording surfaces, the optimum pulse shape may be different for each head. A programmable detector allows each track of each recording surface to be read with its own optimal set of parameters.

A programmable sequence detector may be built using the ACS sharing principles of this invention. In this case, care should be taken to insure that the chosen sharing arrangement is allowed for all possible programmed values of the expected samples.

Multiple-Samples per Sequence-Model State Transition

subset of potential error events for which you would like to preserve the error-handing performance of the full Viterbi demodulator.

For our EPR4 example, we choose to include the set of potential error events whose distance is less than or equal to 2, since 2 is the minimum distance of all closed potential error events and there exist no open potential error events longer than our path memory of distance less than or equal to 2. Using the Euclidean norm, the distance of an error event is the square root of the sum of the squares of the differences in each element between the two codewords comprising the error event. From our example list in step 3, the following 3 potential error events of length 4 are selected:

CODEWORD	STATE SEQUENCE	DISTANCE
0 1 2 1	0 0 1 3 7	
1 2 1 0	0 1 3 7 7	2 (closed)
0 1 2 0	0 0 1 3 6	
1 2 1 0	0 1 3 7 7	1.732 (open)
0 1 2 0	0 0 1 3 6	

handled properly. The longest potential error event at this distance spans six frames of the original single-sample trellis. The same potential error event may span parts of four frames in the double-sample trellis, so for this sharing diagram the path length was taken as P=8, meaning that potential error events up to eight samples long were considered. Since the samples are processed in pairs, only potential error events of even length are considered. Note that the sharing allowed in this case is identical to the sharing allowed when the same sample sequence model is applied one sample at a time.

Both grouping of sequence-model states to share ACS modules and modification of sample sequence models to process multiple samples per transition can be implemented in the same sequence detector. FIGURE 32 is a diagram showing how states from FIGURE 30 may be grouped to share two ACS modules. Each ACS module handles four input paths. Note that there are parallel paths in the figure. For example, if state 011 is currently represented by the ACS module for the left-hand group of states, then branch B3 represents a branch from 011 to 100 with a label 0, -2/1, 0, while branch B4 represents a branch from 011 to 111 with a label 1, 0/0, 0. Each of these branches represents a possible sequence model transition and both must be considered when the surviving path is selected by the ACS module to which they go. FIGURE 32 implies an architecture with which a sequence detector may be implemented for EPR4 signals with d=1 at a processing rate of one-half the sample rate.

FIGURE 33 shows a ten-state modified sample sequence model for partial response signals corresponding to the polynomial (1-D)(1+D)^3 with a minimum run length constraint of d=1. This model is modified from FIGURE 20 in that each sequence model transition in the modified model represents the combination of two consecutive sequence model transitions in the original model of FIGURE 20. The labels in FIGURE 33 indicate pairs of expected samples and pairs of corresponding estimated channel bits.

When the sample sequence model of FIGURE 33 is analyzed to determine allowable sharing of ACS modules, the result is the same as for the single-sample case. For proper resolution of all potential error events of distance less than or equal to the square root of 10, the sharing diagram is as shown in FIGURE 21.

FIGURE 34 is a diagram showing how states from FIGURE 33 may be grouped to share three ACS modules. Each ACS module handles three input paths. In this case, no parallel paths are required. FIGURE 34 implies an architecture with which a sequence detector may be implemented for partial response signals corresponding to the polynomial (1-D)(1+D)^3 with d=1 at a ACS module cycle rate of one-half the sample rate.

FIGURE 35 is a diagram showing how states from FIGURE 33 may be grouped to share five ACS modules. Each ACS module handles two or three input paths, and no parallel paths are required. FIGURE 34 shows how a sequence detector may be implemented for partial response signals corresponding to the polynomial (1-D)(1+D)^3 with d=1 at an ACS module cycle rate of one-half the sample rate. As in the unmodified or single-sample case, the branch labels in this architecture only depend on their state of origin for their signs, allowing partial precomputation of the branch metrics.

FIGURE 36 shows a modified four-state sample sequence model for PR4 signals without run length constraints. This model differs from FIGURE 13 in that each sequence model transition in this model represents the combination of two consecutive sequence model transitions in the original model of FIGURE 13. The labels in FIGURE 36 indicate pairs of expected samples and pairs of corresponding estimated channel bits.

When the sample sequence model of FIGURE 36 is analyzed to determine allowable sharing of ACS modules, the result is the same as for the single-sample case. For proper resolution of all potential error events of distance less than or equal to the square root of 2, the sharing diagram is as shown in FIGURE 14.

FIGURE 37 is a diagram showing how states from FIGURE 36 may be grouped to share two ACS modules. Parallel paths are required, and each ACS module handles four input paths. FIGURE 37 shows how a sequence detector may be implemented for PR4 at an ACS module cycle rate of one-half the sample rate. Note that in the prior art, sequence detectors for PR4 have been built by de-interleaving the sample sequence into two sub-sequences, each of which is demodulated independently using a Viterbi detector for a (1-D) partial response signal. The processing rate in each interleave is one-half the sample rate. This de-interleaved demodulation is possible for PR4 because the PR4 polynomial (1-D)(1+D) = (1 - D^2) contains only even powers of D, and therefore the dependence between samples is restricted to samples in the same interleave. The present invention gives an alternative way to achieve parallelism and reduce the processing rate in PR4 sequence demodulators.

In the claims to follow, the word detector is used in the general sense to denote any sequence detector, whether used for demodulation, decoding or otherwise.

While preferred and alternate embodiments of the present invention have been disclosed and described in detail herein, it will be obvious to those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope thereof.

MATLAB PROGRAM

```
% script file shareacs.m generates a list of codewords for a given state
% machine and identifies all error events (starting in the same state and
% differing in the first transition).  Then it goes through all error
% events, screens them against some specified criteria, and determines
% which states in the state machine can share an ACS unit and path memory
% without compromising the detector's ability to handle the specified set
% of error events.  It can also analyze the allowable sharing under the
% conditions that 2 or more samples are processed per clock cycle.
%
% The state machine is described by a combined transition/input matrix, A,
% The matrix is square, and the i,j element is the expected sample for the
% path from state i to state j.  If there is no path from state i to state
% j, the matrix A should contain a NaN in the i,j position.  A corresponding
% matrix O gives estimated channel bit for each transition specified in A
% (the elements of O that don't correspond to transitions in A are don't
% cares).
%
% Richard T. Behrens, February 1992.
%

format compact
%
% Accept user input to define the problem:
%
clc
A = input('Enter the state transition/input matrix: ');
O = input('Enter the output matrix: ');
nclock = input('Enter the number of samples to process per clock cycle:  ');
Nmax = input('Enter the decoder path length: ');
if (rem(Nmax,nclock)~=0)
   error('Path length must be a multiple of the number of samples per clock.')
end
normtype = input('Enter the desired norm (use 2 for Euclidean): ');
disp('Now specify what types of error events you want to include: ')
disp('     (1) Events with distance <= a given "critical" distance. ')
disp('     (2) Events that can be characterized as dropouts.')
disp(' ')
critdist = input('Enter the "critical" distance:  ');
savedrop = input('Include dropout errors (0/1)?  ');
if savedrop
   dl = input('Max number of dropped out bits to consider: ');
else
   dropout = 0;
end
tellwhy = input('Do you want examples of why each share is forbidden (0/1)? ');

n = length(A);           % number of states
```

```
[T,I,D] = statconv(A,O,nclock);      % convert the state machine to another
format

sharetable = ones(n,n); % begin by assuming all states are shareable
for N = nclock:nclock:Nmax     % for each length of codeword
   disp(['Considering codewords of length ' int2str(N)])
   for i = 1:n           % for each starting state ...
      disp(['Working on error events that start in state ' int2str(i)])
      tidx = find(T(:,1)==i);       % find all transitions from state i
      stateseqs = T(tidx,:)';       % keep track of state sequences ...
      cws = I(tidx,:)';             % and codewords ...
      outputs = D(tidx,:)';         % and channel bits.
      [dummy,M] = size(cws);
      for k = (2*nclock):nclock:N
         newstateseqs = [];
         newcws = [];
         newoutputs = [];
         for j = 1:M
            tidx = find(T(:,1)==stateseqs(k/nclock,j));
            stateseq = stateseqs(:,j)*ones(1,length(tidx));
            newstateseqs = [newstateseqs [stateseq; T(tidx,2)']];
            cw = cws(:,j)*ones(1,length(tidx));
            newcws = [newcws [cw; I(tidx,:)']];
            output = outputs(:,j)*ones(1,length(tidx));
            newoutputs = [newoutputs [output; D(tidx,:)']];
         end
         stateseqs = newstateseqs;
         cws = newcws;
         outputs = newoutputs;
        [dummy,M] = size(cws);   % codewords of length k from state i
      end

      % now we have a list of all M codewords that start from state i,
      % and we can investigate all the error events that begin in state i.

      for j = 1:(M-1)
         for k = (j+1):M
            if (stateseqs(2,j)~=stateseqs(2,k))   % if pair is an error event
                  dist = norm(cws(:,j)-cws(:,k),normtype);   % find distance
                  if savedrop      % check for dropout/dropin event
                     dropout = isdrop(outputs(:,j),outputs(:,k),dl);
                     dropout = dropout|isdrop(outputs(:,k),outputs(:,j),dl);
                  end
                  if (dist<critdist)|dropout    % if an event of interest
                     sti = 2;   % start at the beginning of the error event
                     while (sti<=(N/nclock+1))   % until the event closes
                        if (stateseqs(sti,j)~=stateseqs(sti,k))
                           stj = stateseqs(sti,j);
                           stk = stateseqs(sti,k);
                           if sharetable(stj,stk)
                              sharetable(stj,stk) = 0; % share not allowed
                              disp(['Marking nonshareable state pair: ' ...
                                     int2str(stj) ', ' int2str(stk)])
                              if tellwhy
                                 disp('An error event needing this pair: ')
                                 disp('Codewords: ')
                                 disp([cws(:,j) cws(:,k)]')
                                 disp('State sequences: ')
                                 disp([stateseqs(:,j) stateseqs(:,k)]')
                                 disp('Outputs: ')
                                 disp([outputs(:,j) outputs(:,k)]')
                                 disp(['Distance = ' num2str(dist)])
                                 if dropout
                                    disp('This is a dropout event.')
                                 end
                                 pause
                                 clc
                              end
                           end
                           sti = sti + 1;
                        else
                           break
                        end
                     end
                  end
               end
            end
         end
      end
   end
end

clc
sharetable = sharetable&(sharetable');
disp('Sharing table: ')
disp([nan 1:n; [(1:n)' sharetable]])

======================================================================
function dropout = isdrop(out1,out2,dl)
% dropout = isdrop(out1,out2,dl) returns a 1 if the error between
% binary sequences out1 and out2 can be characterized as a dropout,
% otherwise returns a 0.  A dropout is indicated when:
%          (1) every 0 in out1 is 0 in out2 (no drop in or shift)
%  and     (2) at least one 1 in out1 is 0 in out2  (dropout)
%  and     (3) not more than dl 1s in out1 are 0 in out2 (not too many)
%
% If dl is not supplied, it is taken as infinity.
%
% Richard T. Behrens, February 1992.
%
if nargin<3
   dl = inf;
end
numdrop = sum((out1==1)&(out2==0));
dropout = (~any((out1==0)&(out2==1))) & (numdrop>0) & (numdrop<=dl);

======================================================================
function [T,I,D] = statconv(A,O,n)
% [T,I,D] = statconv(A,O,n) converts a state machine from the input format
% used by shareacs.m (and described therein) to a format suitable to
% describe state machines that process n samples per clock cycle.  The
% new format consists of three matrices.  T is a list of transitions, with
% one row for every transition path in the machine, and two columns
% containing the state numbers that the transition path comes from and
% goes to.  I is a list of input labels (expected samples), with one row
% for every row in T, and one column for every sample processed in one
% clock cycle (n).  D is a list of output labels (channel bits) in the
% same format as I.
%
% Richard T. Behrens, February 1992.
%
ns = length(A);          % number of states

T = []; I = []; D = [];
for i = 1:ns             % for each starting state
   trans = find(~isnan(A(i,:)));    % find valid transitions
   stateseqs = trans;               % keep track of state sequences ...
   cws = A(i,trans);                % and input labels ...
   outputs = O(i,trans);            % and output labels.
   M = length(cws);
   for k = 2:n
      newstateseqs = [];
      newcws = [];
      newoutputs = [];
      for j = 1:M
         trans = find(~isnan(A(stateseqs(k-1,j),:)));
         stateseq = stateseqs(:,j)*ones(1,length(trans));
         newstateseqs = [newstateseqs [stateseq; trans]];
         cw = cws(:,j)*ones(1,length(trans));
         newcws = [newcws [cw; A(stateseq(k-1),trans)]];
         output = outputs(:,j)*ones(1,length(trans));
         newoutputs = [newoutputs [output; O(stateseq(k-1),trans)]];
      end
      stateseqs = newstateseqs;
      cws = newcws;
      outputs = newoutputs;
      [dummy,M] = size(cws);   % number of codewords of length k from state i
   end

   % add the length n codewords to the transition path list
   I = [I; cws'];
   T = [T; [i*ones(M,1) stateseqs(n,:)']];
   D = [D; outputs'];
end

=======================================================================
```

Sometimes the required sample rate and corresponding channel bit rate exceed the maximum ACS module cycle rate, i.e. the rate at which a particular implementation of an ACS module can accept new actual sample values and select surviving paths. When this occurs, a means is needed to restructure the computations so that more processing may be done in parallel. This can be accomplished in a sequence detector by beginning with a suitably modified sample sequence model.

Consider the case where a doubling of the rate at which an ACS module processes read-signal samples is sufficient. In this case, the sample sequence model is modified by considering each pair of possible consecutive sequence model transitions as a single state transition in the modified sample sequence model. All sequence-model transitions of the modified sample sequence model represent all possible consecutive pairs of transitions from the original sample sequence model. The modified detector accepts a pair of samples of the analog read signal, generates branch metrics by comparing the received sample pair to expected sample pairs for each branch, adds the compound branch metrics to previous path metrics and selects the surviving paths. In effect, the sample sequence model is modified to accept and process multiple samples per state transition, and the fundamental frame of the corresponding trellis is replaced by a modified fundamental frame whose branches represent all paths through two consecutive fundamental frames of the original trellis.

The advantage of this modification is that the add, compare, select process can operate at one half of the sample rate. The cost of this modification is that the compound branch metrics are more complex to compute and that there may, in general, be more paths entering each ACS module. The comparisons among the greater number of path metrics may be carried out in parallel, as may the computation of the compound branch metrics. As will be clear to one skilled in the art, an analogous modification can be used to process 3 or more samples in parallel per cycle of the ACS module.

FIGURE 30 shows a six-state modified sample sequence model for EPR4 with a minimum run length constraint of d=1. This model differs from FIGURE 3 in that each sequence-model transition in this model represents the combination of two consecutive sequence model transitions in the unmodified sample sequence model of FIGURE 3. The labels in FIGURE 30 indicate pairs of expected samples and pairs of corresponding estimated channel bits.

FIGURE 31 is a sharing diagram obtained when we require that all potential error events of distance less than or equal to 2 be

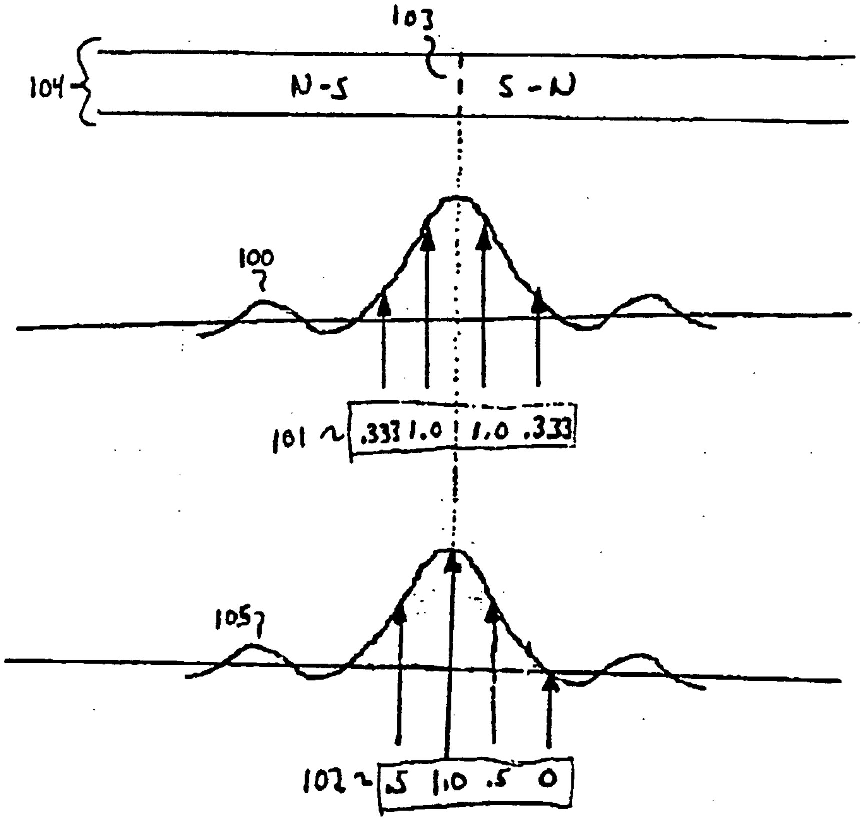


FIG. 1
(PRIOR ART)

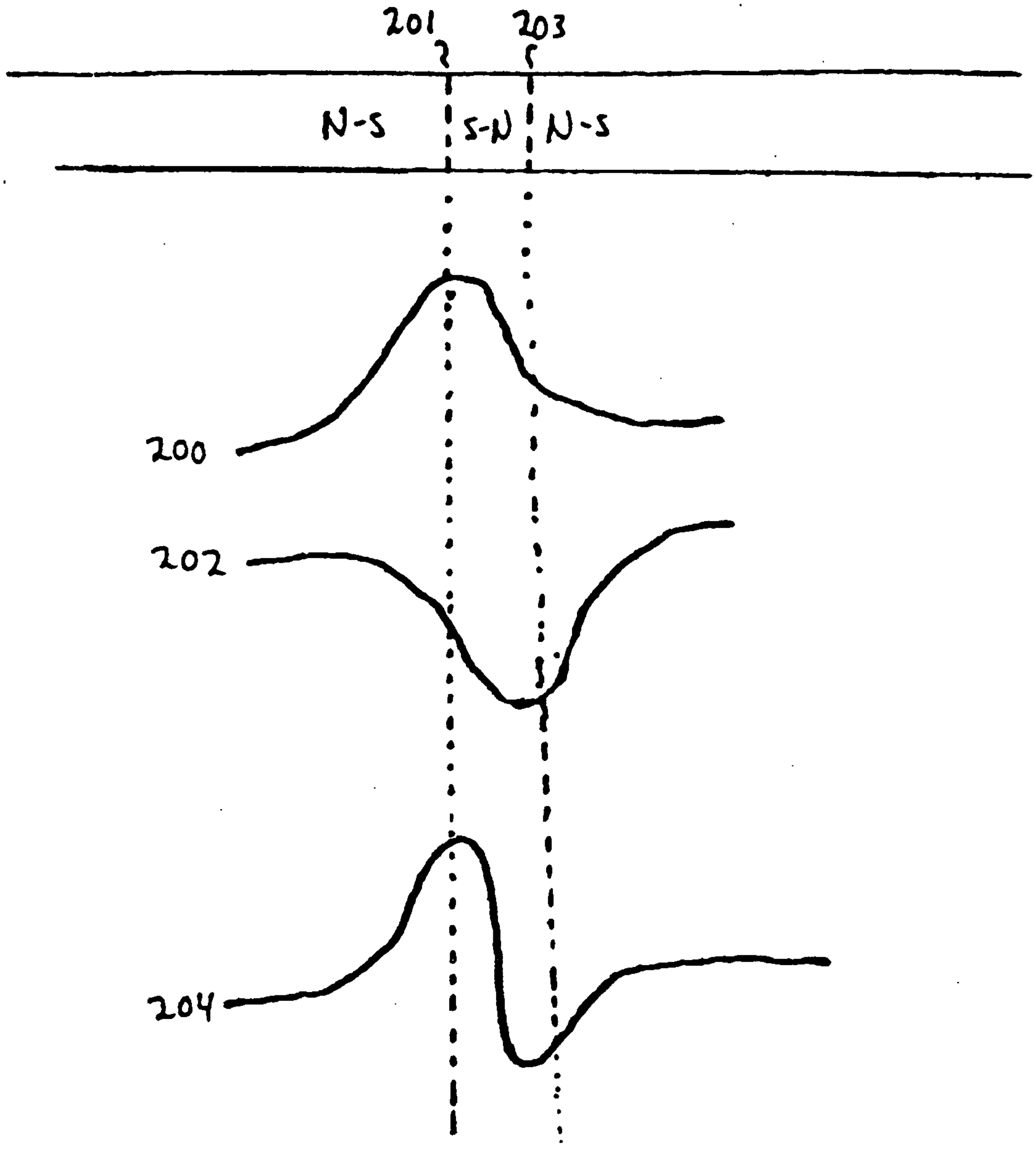


FIG. 2 (PRIOR ART)

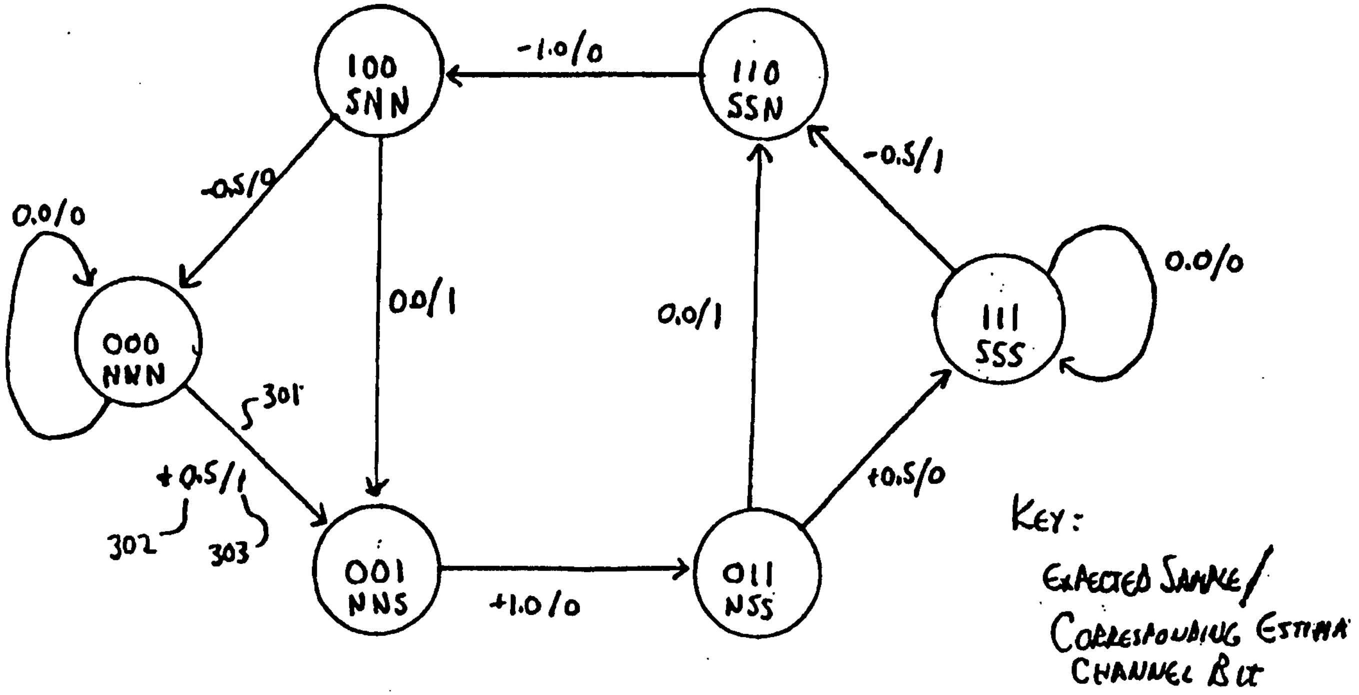
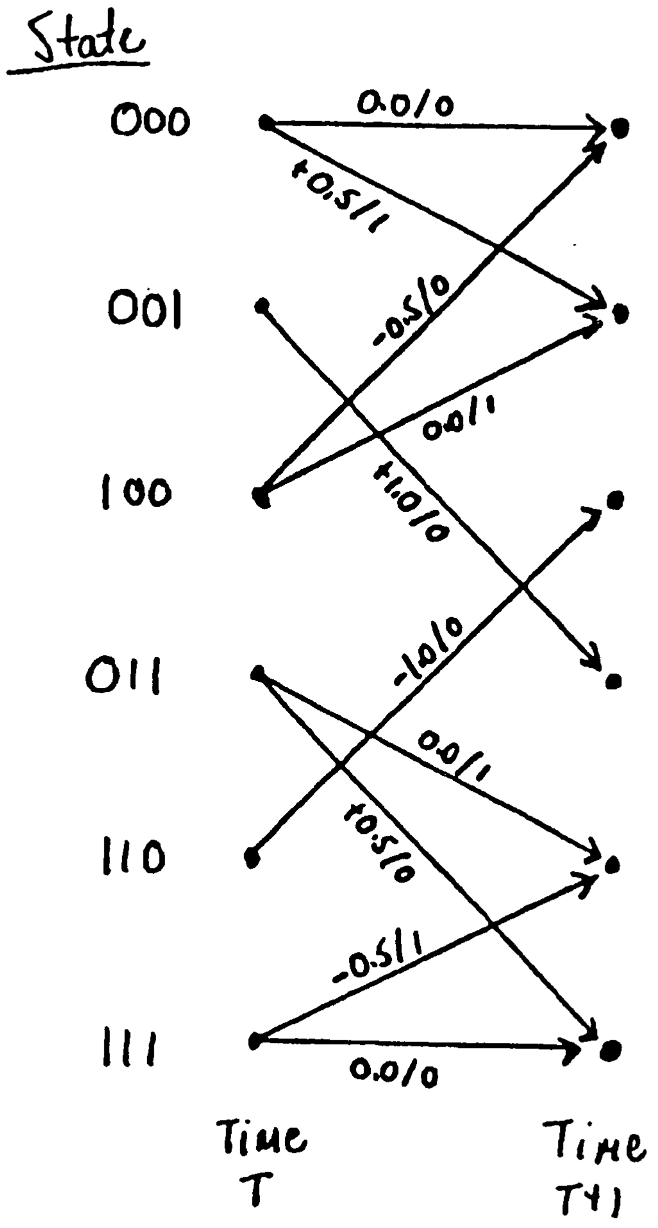


FIG. 3
(PRIOR ART)



Key: Expected Sample / Corresponding Estimated Channel Bit

FIG. 4
(PRIOR ART)

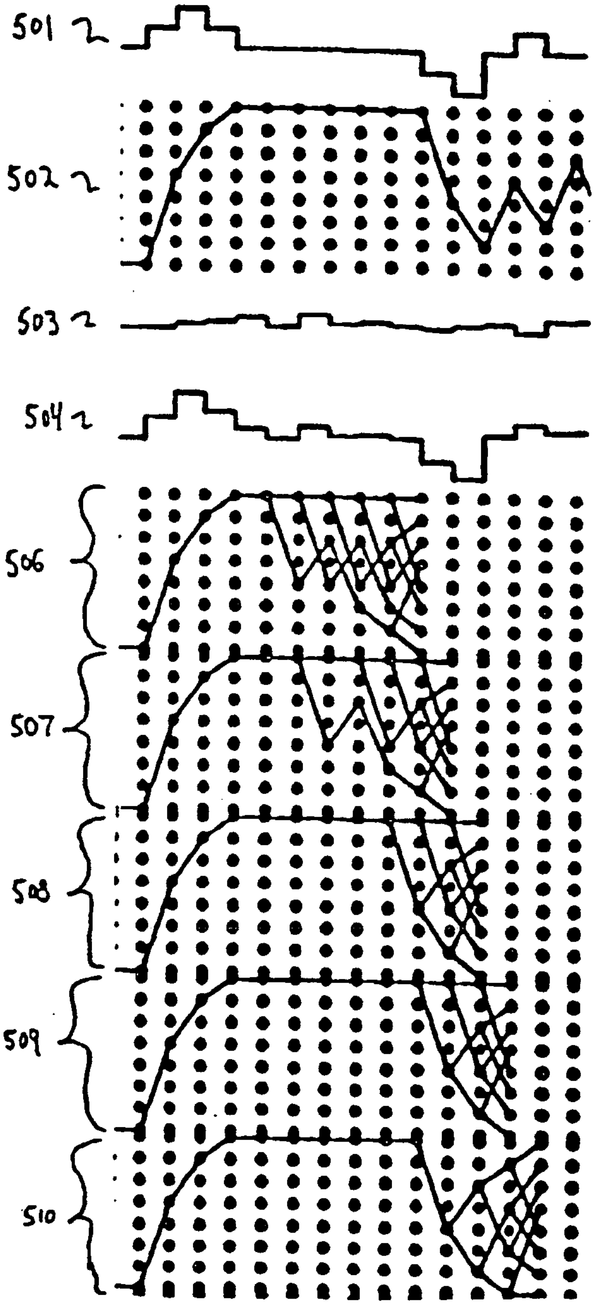


FIG. 5 (PRIOR ART)

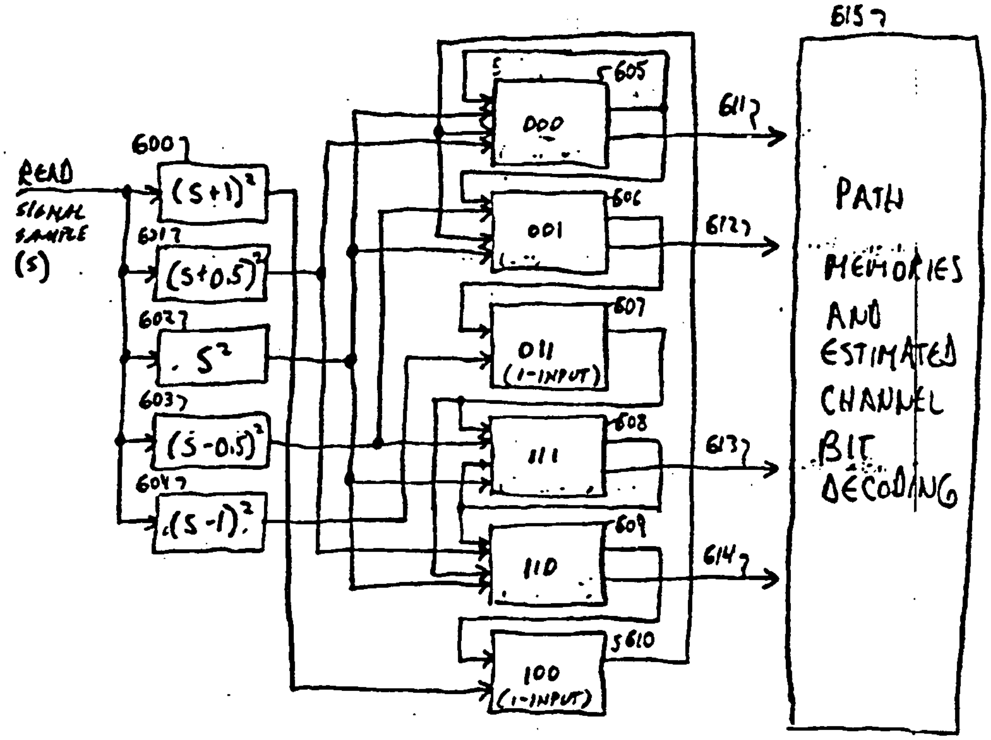


FIG. 6
(PRIOR ART)

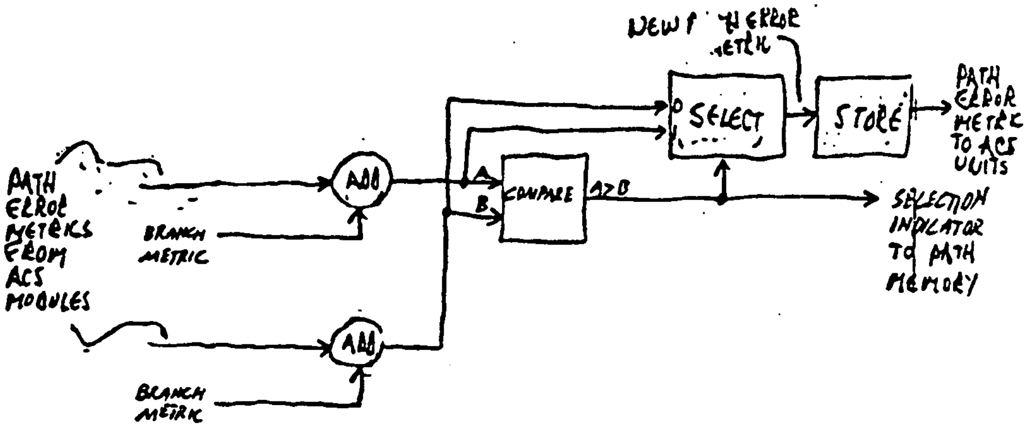

NEW ERROR METRIC
SELECT
STORE
PATH ERROR METRIC TO ACS UNITS
PATH ERROR METRICS FROM ACS MODULES
BRANCH METRIC
ADD
A
B
COMPARE
A≥B
SELECTION INDICATOR TO PATH MEMORY
ADD
BRANCH METRIC

FIG. 7
(PRIOR ART)

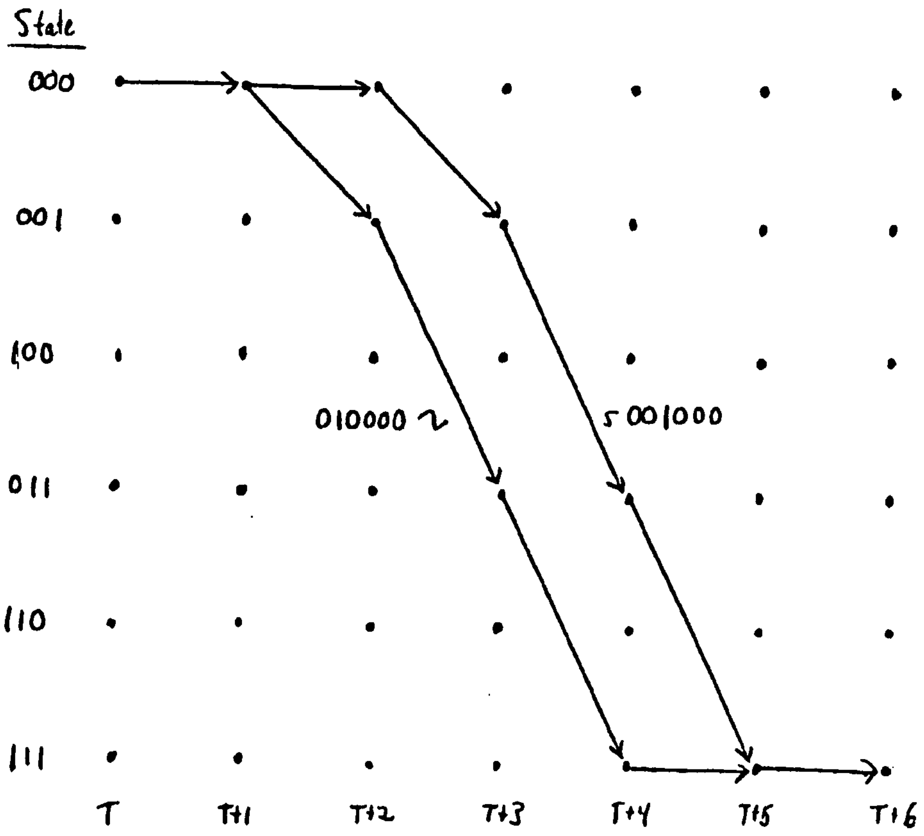
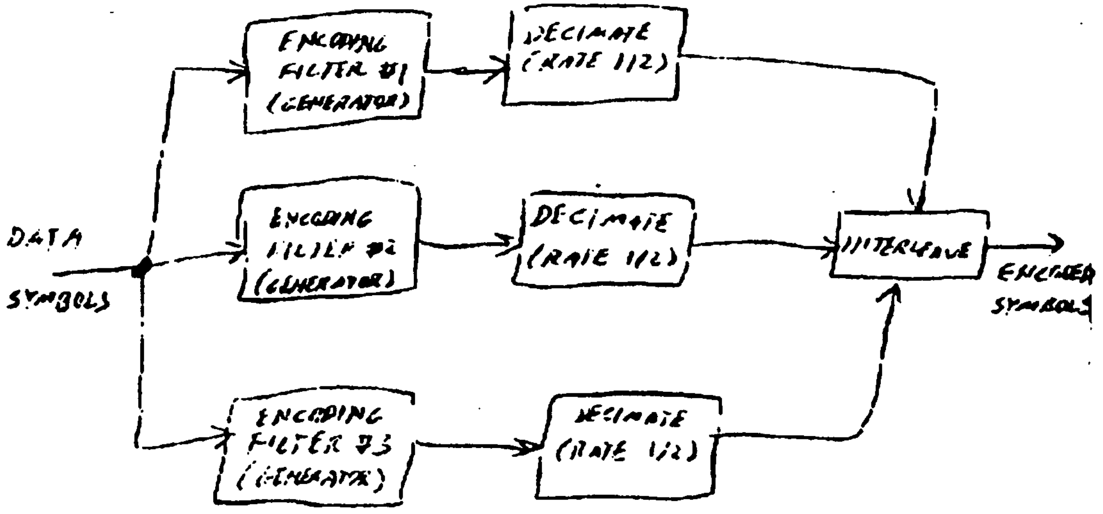


FIG. 8
(PRIOR ART)

FIGURE 9


DATA
SYMBOLS
ENCODING
FILTER #1
(GENERATOR)
ENCODING
FILTER #2
(GENERATOR)
ENCODING
FILTER #3
(GENERATOR)
DECIMATE
(RATE 1/2)
DECIMATE
(RATE 1/2)
DECIMATE
(RATE 1/2)
INTERLEAVE
ENCODED
SYMBOLS

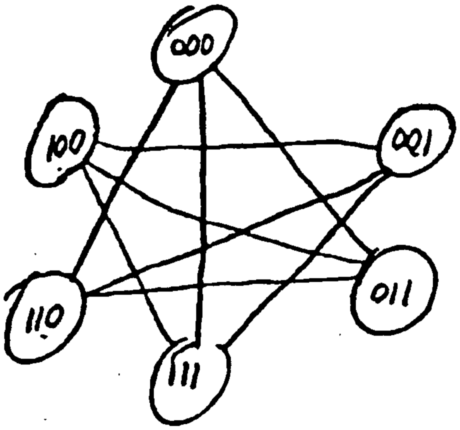


FIG. 10

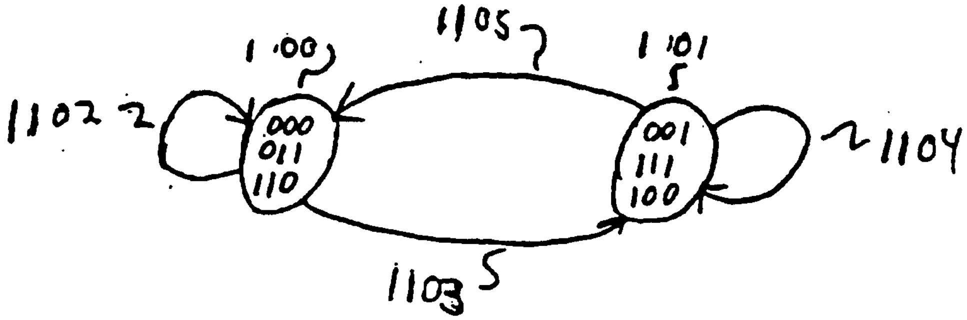


FIG. 11

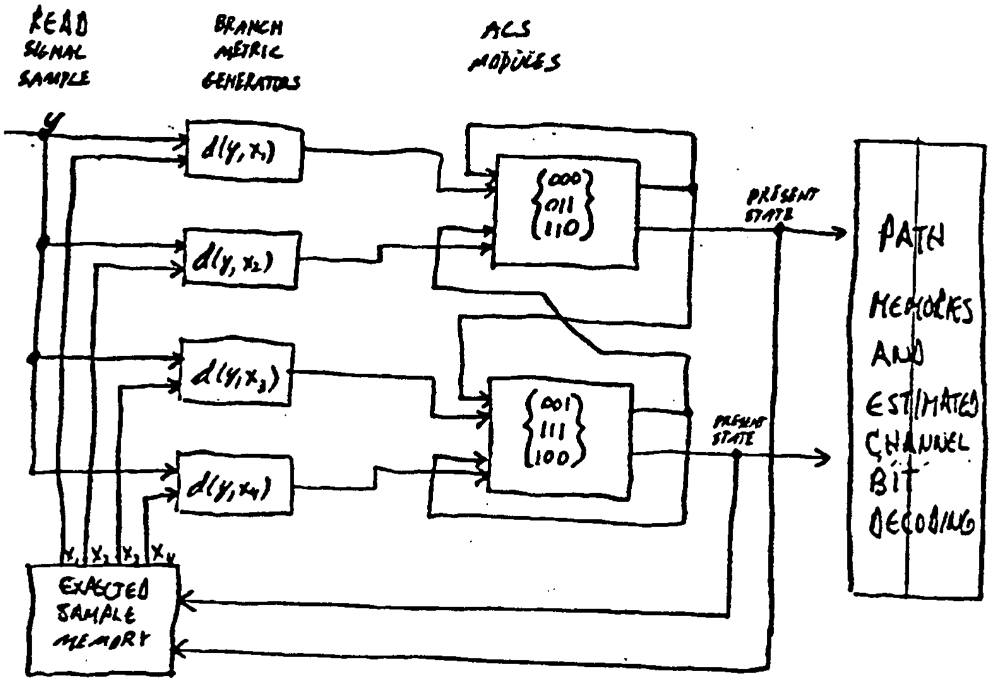

READ SIGNAL SAMPLE
BRANCH METRIC GENERATORS
ACS MODULES
y
d(y, x1)
d(y, x2)
d(y, x3)
d(y, x4)
{000 011 110}
{001 111 100}
PRESENT STATE
PRESENT STATE
PATH MEMORIES AND ESTIMATED CHANNEL BIT DECODING
x1 x2 x3 x4
EXPECTED SAMPLE MEMORY

FIG. 12

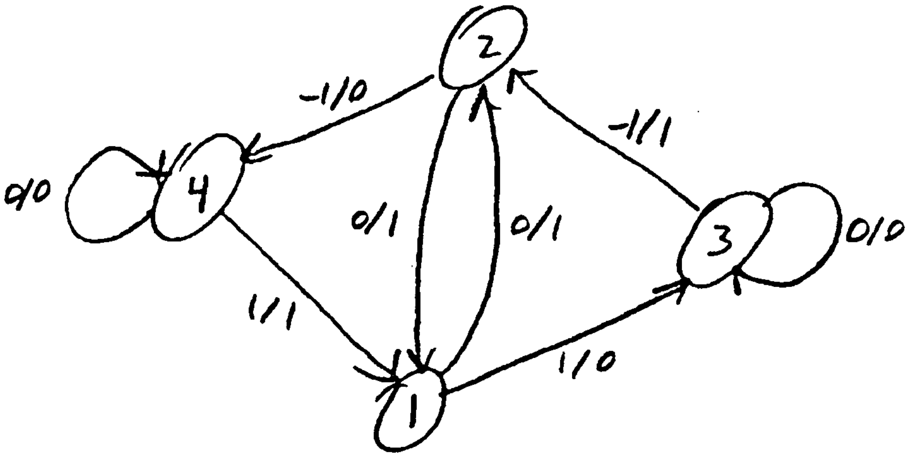


FIG. 13

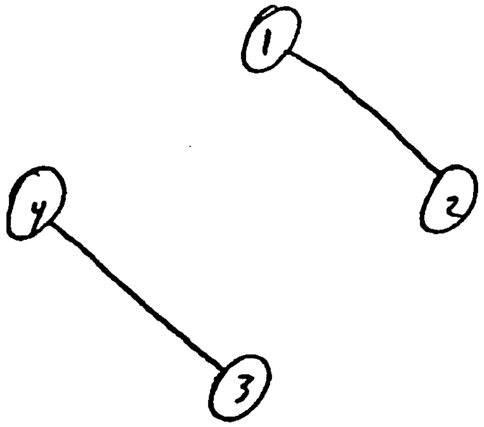


FIG. 14

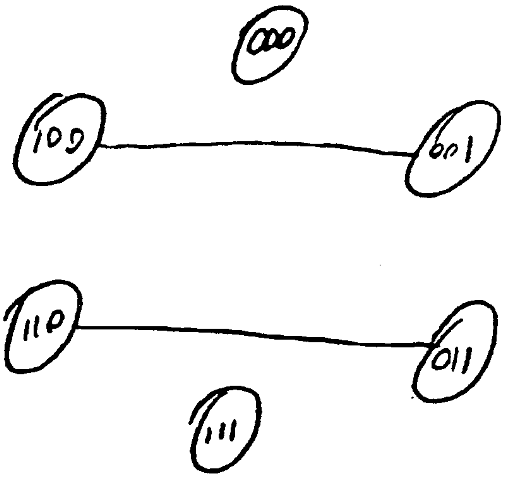


FIG. 15

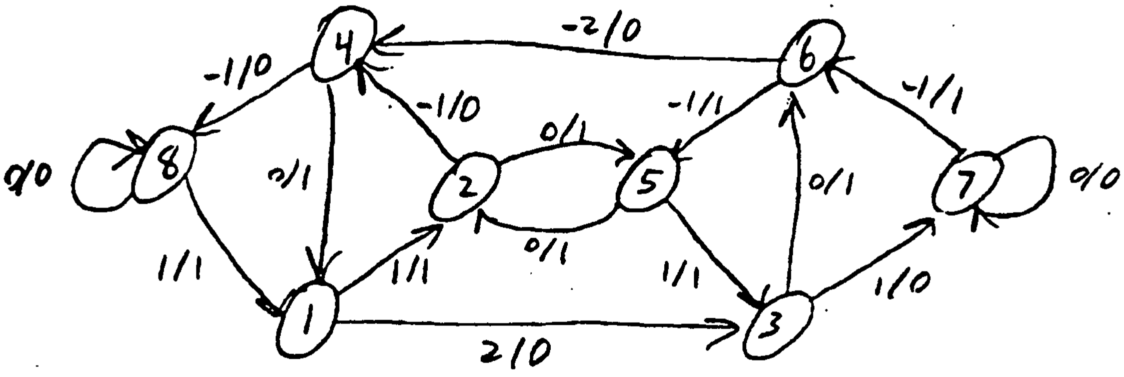


FIG. 16

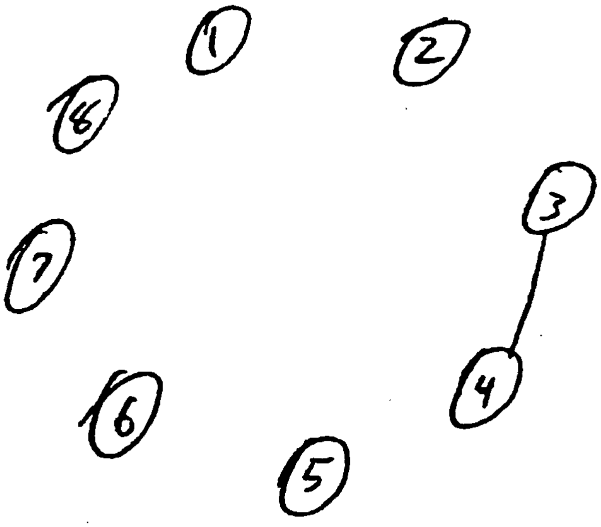


FIG. 17

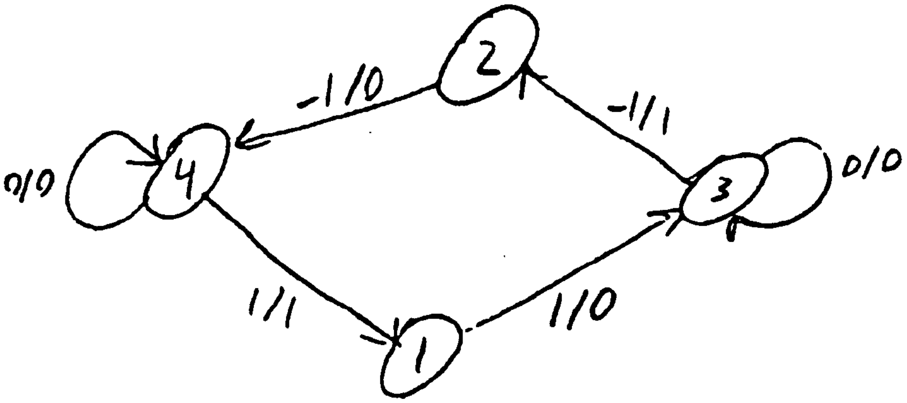


FIG. 18

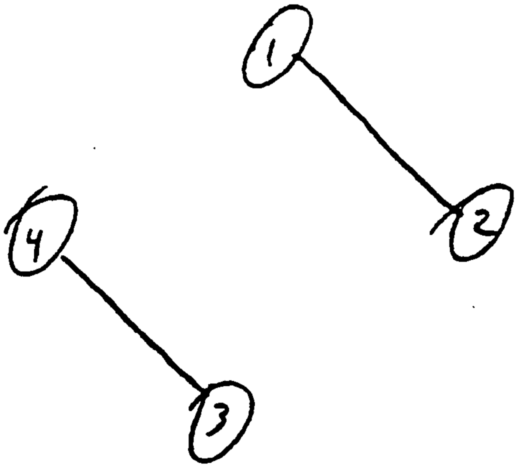


FIG. 19

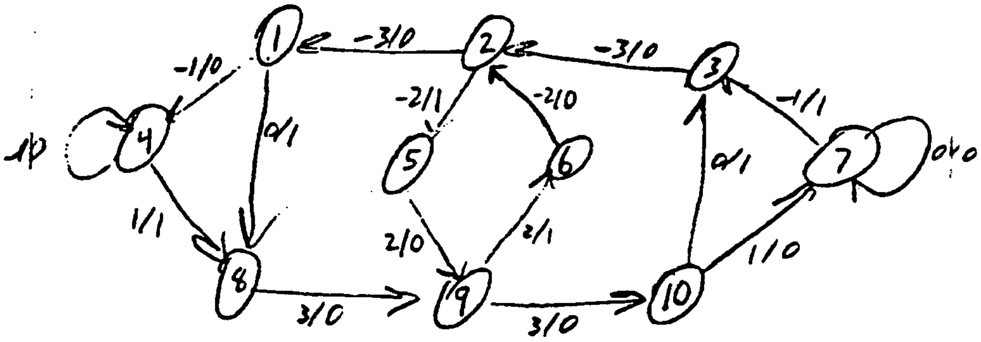


FIG. 20

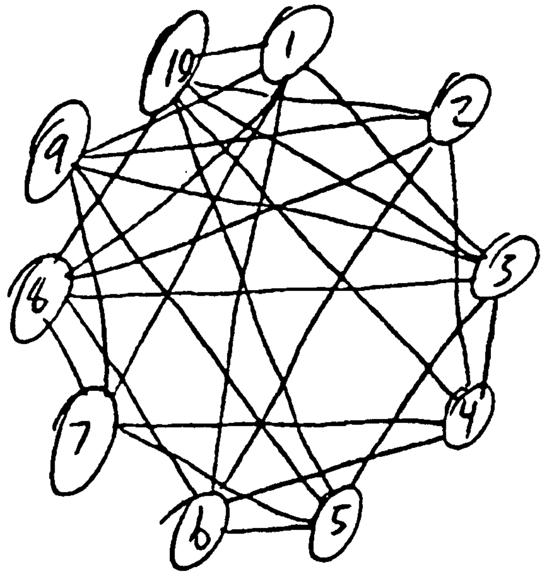


FIG. 21

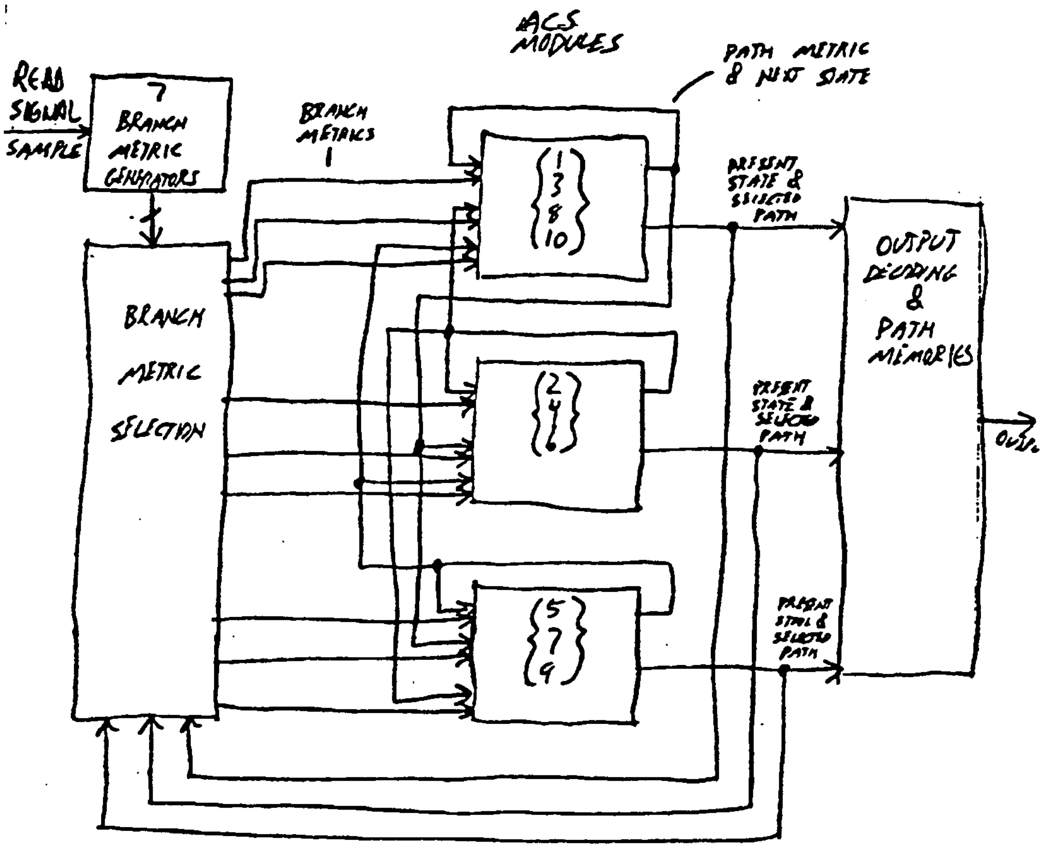
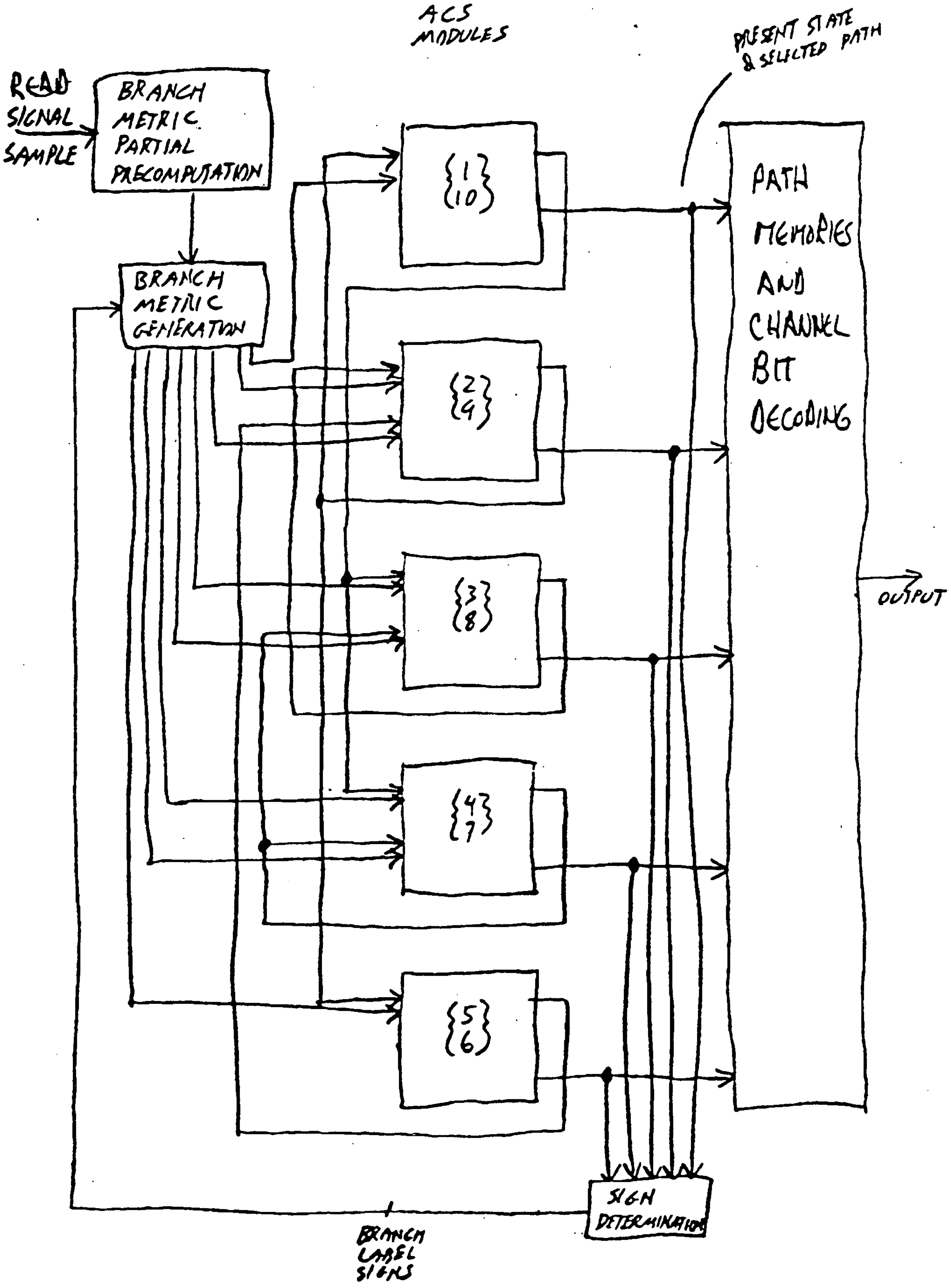

ACS MODULES
PATH METRIC & NEXT STATE
READ SIGNAL SAMPLE
7 BRANCH METRIC GENERATORS
BRANCH METRICS
1
{1 3 8 10}
PRESENT STATE & SELECTED PATH
OUTPUT DECODING & PATH MEMORIES
BRANCH METRIC SELECTION
{2 4 6}
PRESENT STATE & SELECTED PATH
OUTP
{5 7 9}
PRESENT STATE & SELECTED PATH

FIG. 22


ACS MODULES
PRESENT STATE & SELECTED PATH
READ SIGNAL SAMPLE
BRANCH METRIC PARTIAL PRECOMPUTATION
BRANCH METRIC GENERATION
{1 10}
{2 9}
{3 8}
{4 7}
{5 6}
PATH MEMORIES AND CHANNEL BIT DECODING
OUTPUT
SIGN DETERMINATION
BRANCH LABEL SIGNS

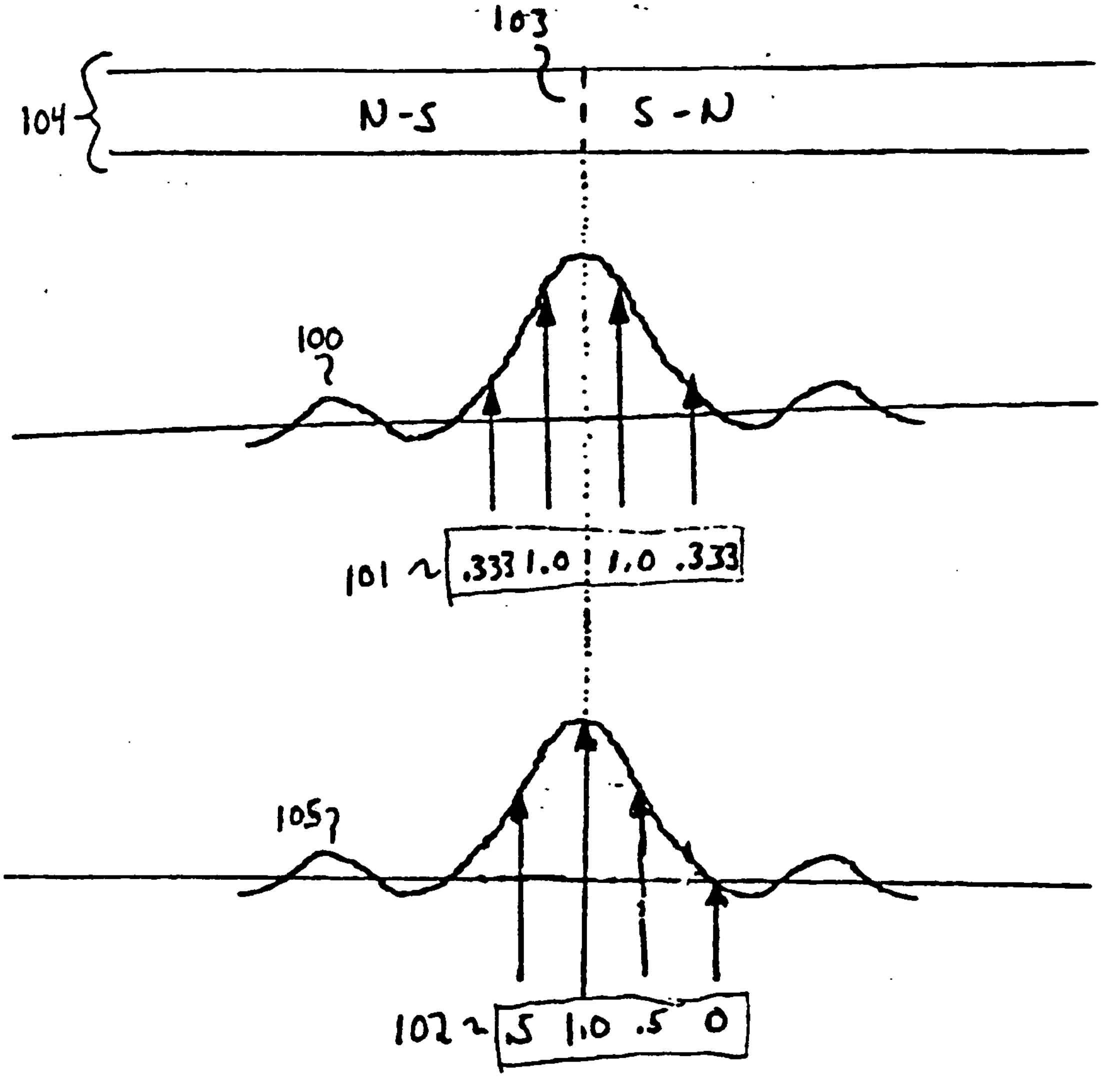


FIG. 1
(PRIOR ART)

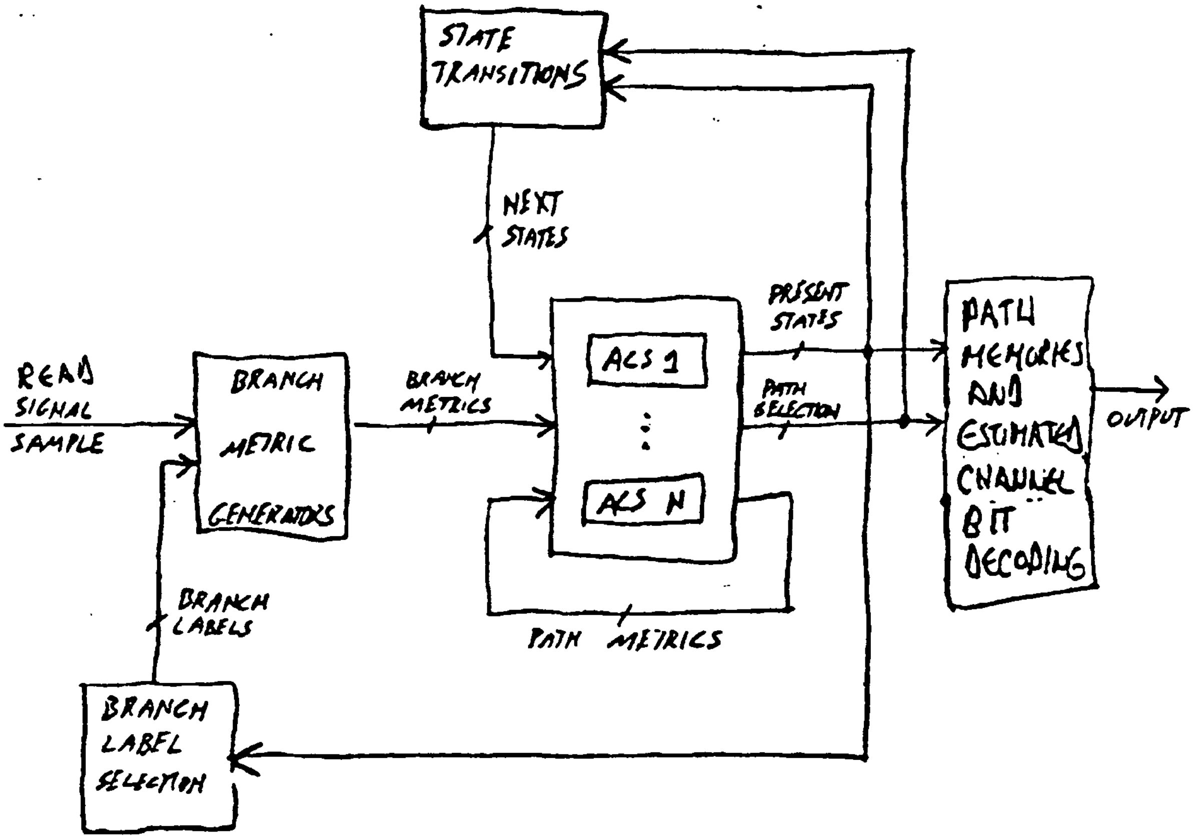

STATE TRANSITIONS
NEXT STATES
PRESENT STATES
PATH SELECTION
PATH MEMORIES AND ESTIMATED CHANNEL BIT DECODING
OUTPUT
READ SIGNAL SAMPLE
BRANCH METRIC GENERATORS
BRANCH METRICS
ACS 1
ACS N
PATH METRICS
BRANCH LABELS
BRANCH LABEL SELECTION

FIG. 25

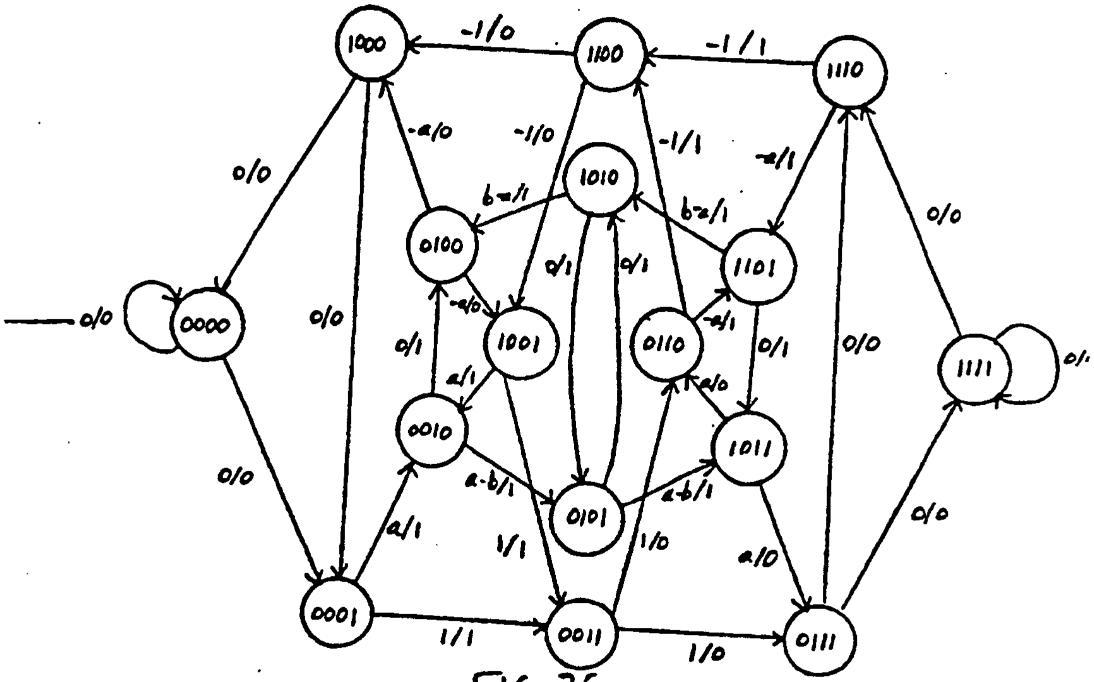

1000
-1/0
1100
-1/1
1110
0/0
-a/0
-1/0
1010
-1/1
-a/1
b-a/1
b-a/1
0/0
0100
1101
0/0
0/0
0000
0/0
-a/0
1001
0/1
0/1
0110
-a/1
0/1
0/0
1111
0/
0/1
a/1
a/0
0010
1011
0/0
a-b/1
0101
a-b/1
0/0
a/1
1/1
1/0
a/0
0001
0011
0111
1/1
1/0

FIG. 26

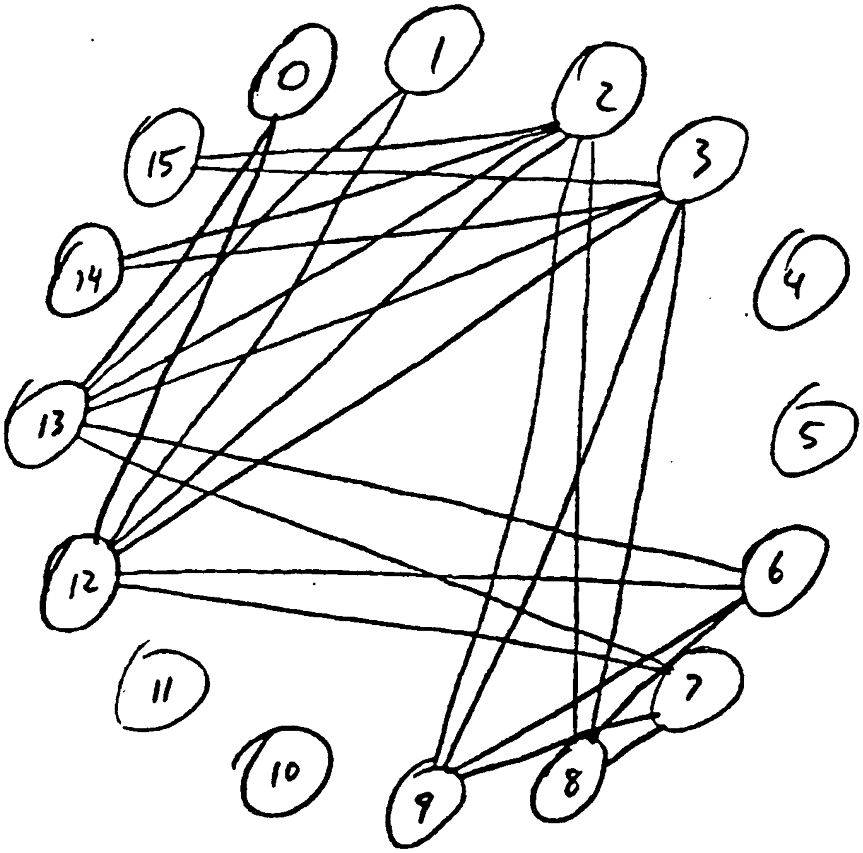


FIG. 27

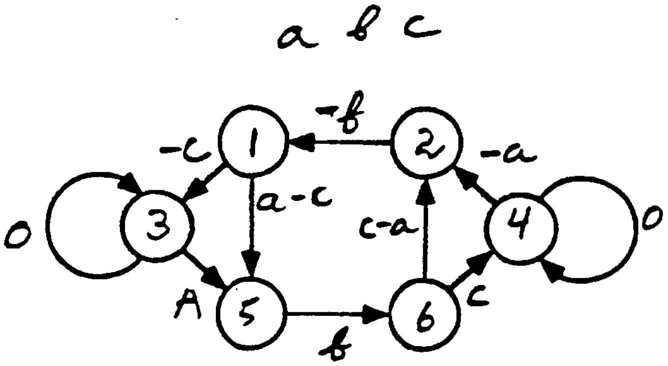


Fig. 28

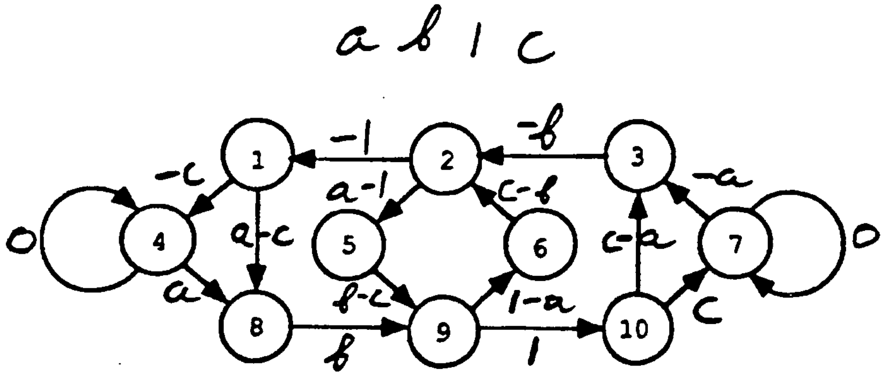


Fig. 29

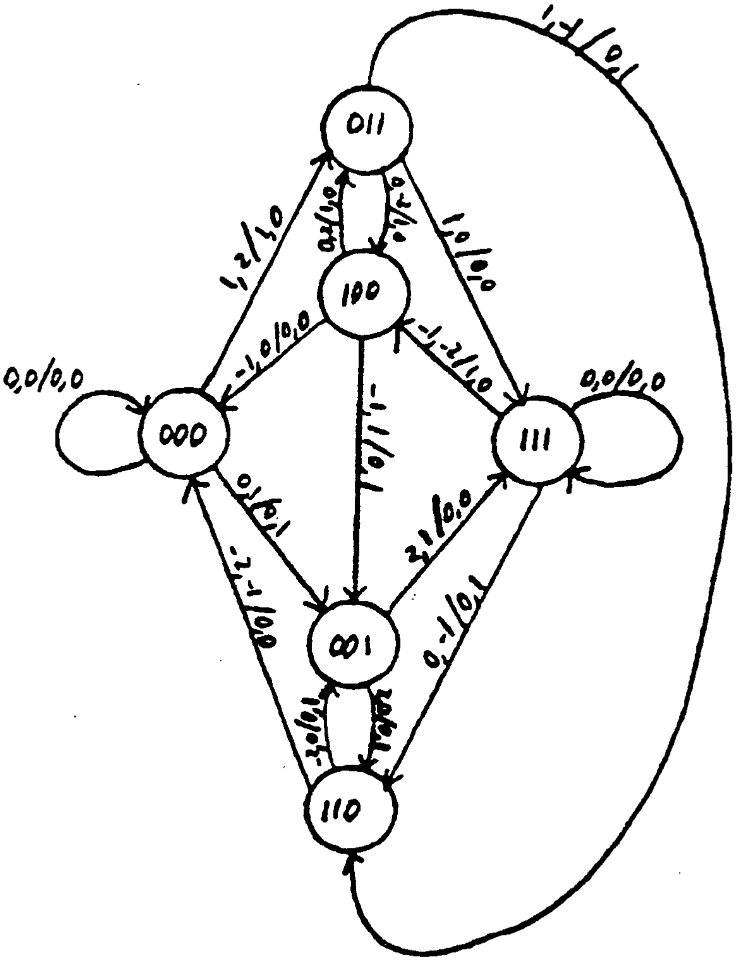

1,-1/0,1
011
1,2/1,0
0,2/1,0
0,-2/1,0
1,0/0,0
100
-1,0/0,0
-1,-2/1,0
0,0/0,0
000
0,0/0,0
111
-1,1/0,1
0,1/0,1
2,1/0,0
-2,-1/0,0
0,-1/0,1
001
-3,0/0,1
3,0/0,1
110

FIGURE 30

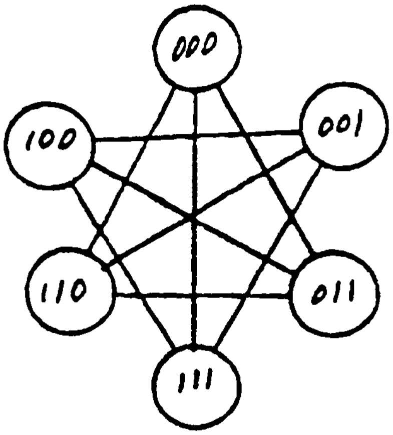


FIGURE 31

(OR SIMPLY REFER AGAIN TO FIGURE 10, WHICH IS IDENTICAL)

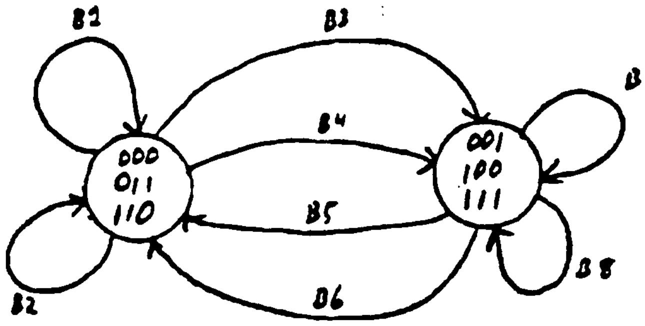


FIGURE 32

BRANCH	PRESENT STATE	LABEL	NEXT STATE
B1	000	0,0/0,0	000
	011	1,-1/0,1	110
	110	-2,-1/0,0	000
B2	000	1,2/1,0	011
B3	000	0,1/0,1	001
	011	0,-2/1,0	100
	110	-2,0/0,1	001
B4	011	1,0/0,0	111
B5	100	-1,0/0,0	000
B6	001	2,0/0,1	110
	100	0,2/1,0	011
	111	0,-1/0,1	110
B7	111	-1,-2/1,0	100
B8	001	2,1/0,0	111
	100	-1,1/0,1	001
	111	0,0/0,0	111

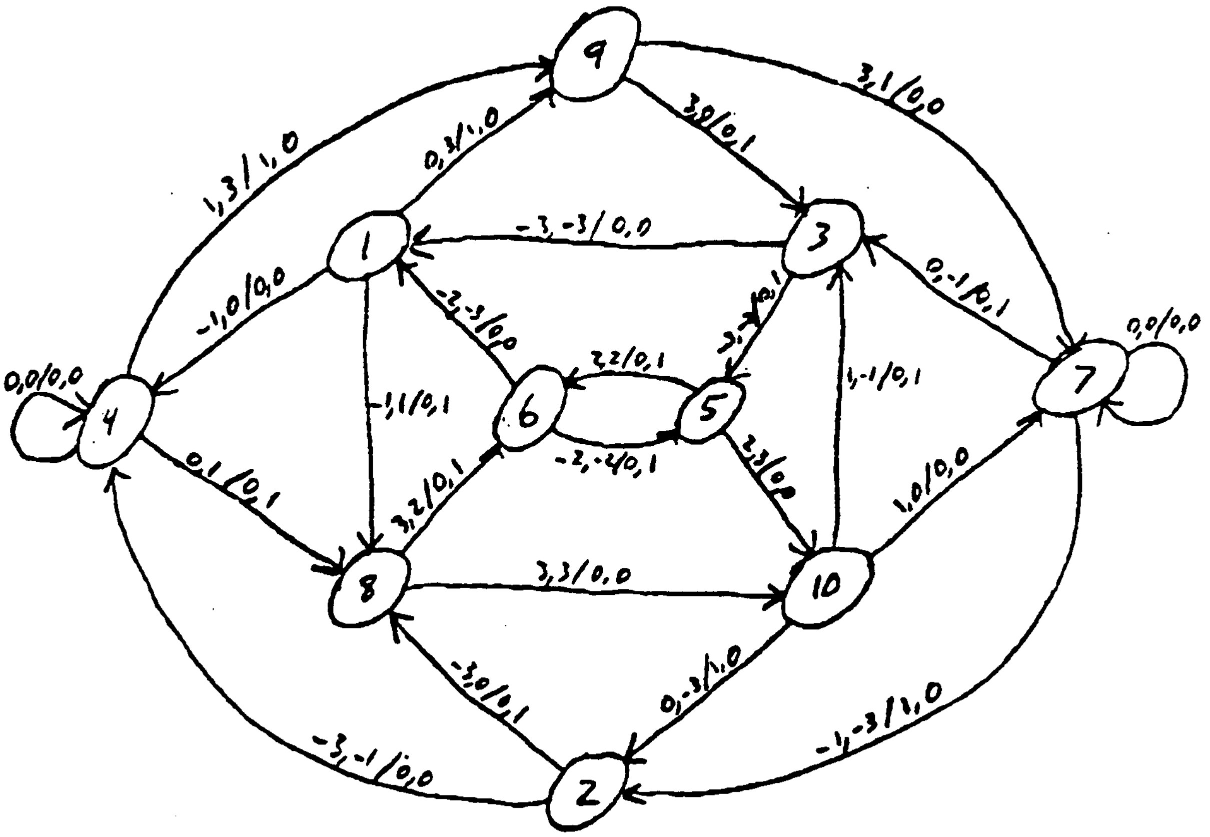

9
1,3/1,0
0,3/1,0
3,0/0,1
3,1/0,0
1
-3,-3/0,0
3
-1,0/0,0
-2,-3/0,0
3,-/0,1
0,-1/0,1
0,0/0,0
0,0/0,0
4
-1,1/0,1
6
2,2/0,1
5
1,-1/0,1
7
-2,-4/0,1
0,1/0,1
3,2/0,1
2,3/0,0
1,0/0,0
8
3,3/0,0
10
-3,0/0,1
0,-3/1,0
-1,-3/1,0
-3,-1/0,0
2

FIGURE 33

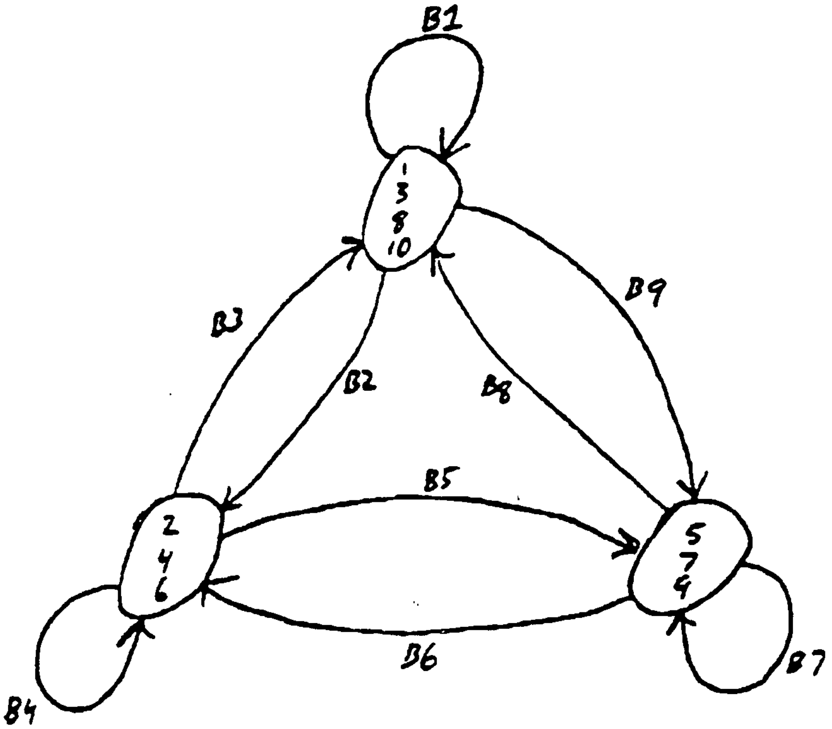


FIGURE 34

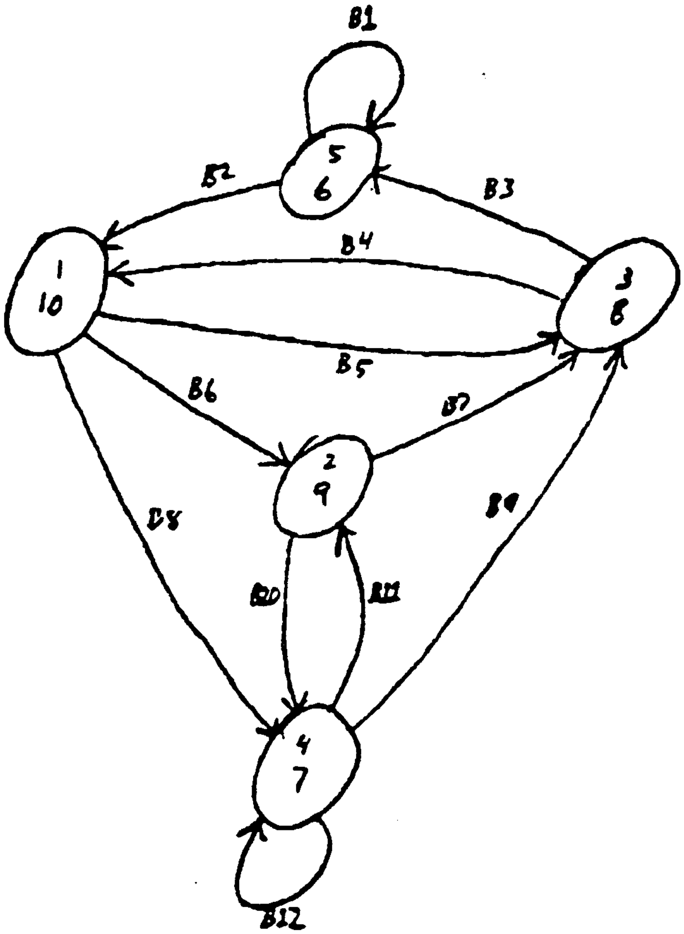


FIGURE 35

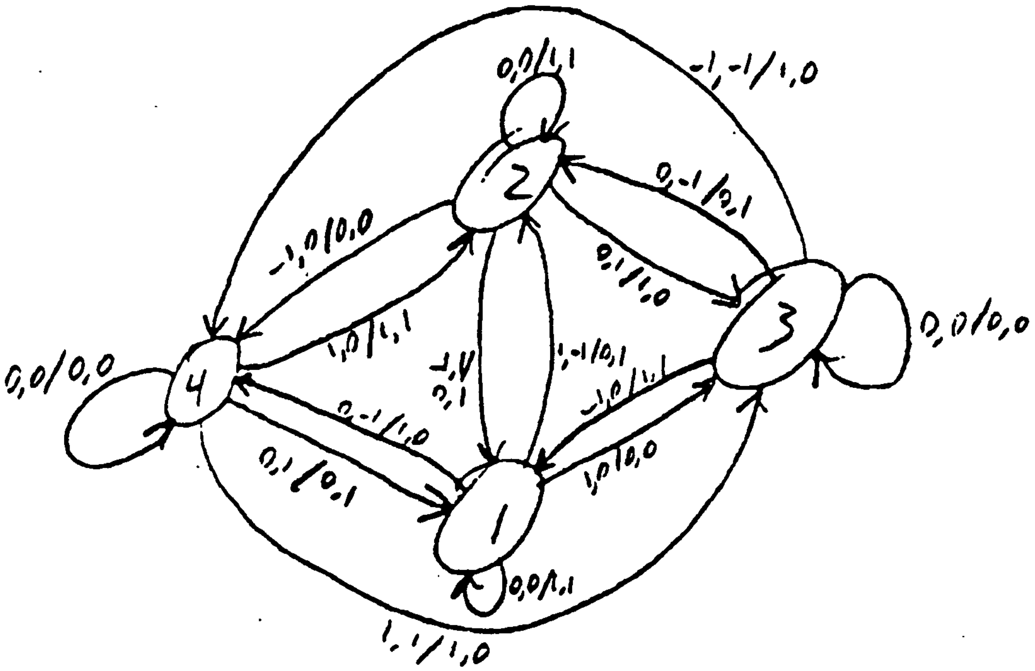

0,0/1,1
-1,-1/1,0
2
0,-1/0,1
-1,0/0,0
0,1/1,0
3
0,0/0,0
1,0/1,1
0,0/0,0
4
-1,1/0,1
1,-1/0,1
-1,0/1,1
0,-1/1,0
1,0/0,0
0,1/0,1
1
0,0/1,1
1,1/1,0

FIGURE 36

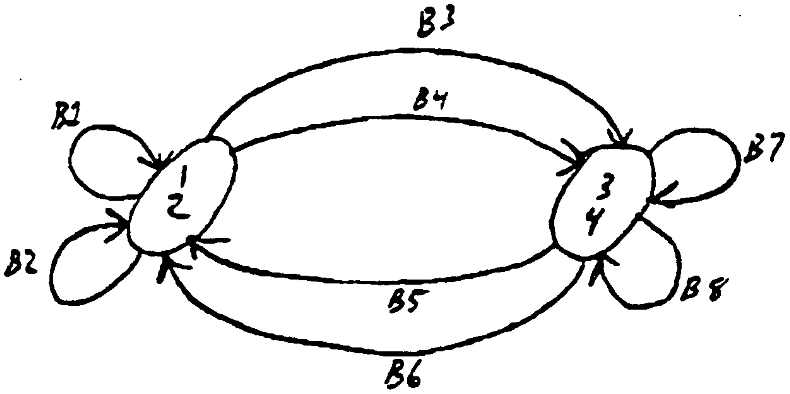


FIGURE 37

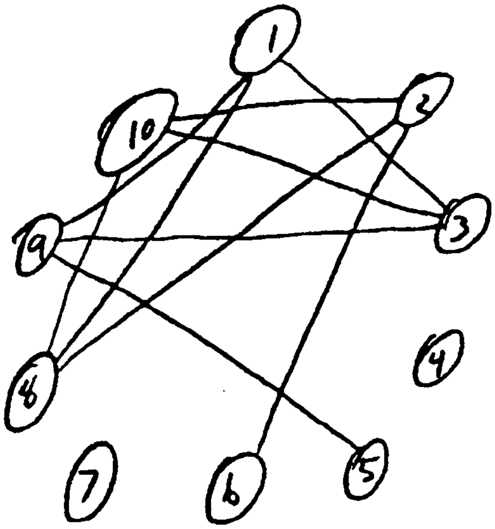


FIG. 24

What is claimed is:

1. An integrated circuit synchronous read channel for receiving digitized read signals representing samples of a read signal of a magnetic storage device and recovering digital data represented thereby comprising:

a transition detector for detecting amplitude pulses of the digitized read signals indicative of storage media transitions and for generating an output signal representative of the amplitude pulses, said output signal being a sequence of binary digital signals;

timing recovery circuitry responsive to the digitized read signal and the output signal of the transition detector to provide a timing control signal for controlling the timing of digitized samples of the read signal;

a sequence detector responsive to the digitized read signals for receiving as stream of the digitized read signals and determining a corresponding sequence of binary digital signals likely to be represented thereby, said sequence detector including a path memory means for constructing sequences of binary digital signals, a comparison means for comparing the received read signal to a set of predetermined ideal read signals, and a selection means for selecting one of said set of ideal read signals which resembles the received read signal and for selecting a sequence of binary digital signals from the path memory means which corresponds to the selected ideal read signal;

an RLL (d,k) decoder for providing a run length limited decoded output by decoding the selected sequence of binary digital signals from the sequence detector or for providing a run length limited decoded output by decoding the sequence of binary digital signals from the transition detector;

control means for directing either the sequence of binary digital signals from the sequence detector or the sequence of binary digital signals from the transition detector to the RLL (d,k) decoder.

2. The integrated circuit synchronous read channel of claim 1 further comprised of digital pulse shaping filter circuitry for modifying the digitized read signals prior to receipt thereof by at least one of (i) the sequence detector, (ii) the transition detector and (iii) the timing recovery circuitry.

3. The integrated circuit synchronous read channel of claim 2 further comprised of delay means for delaying coupling of the digitized read signals to the transition detector or to the timing reciver circuitry to match a delay in coupling the digitized read signals to the timing recovery circuitry or the transition detector, respectively, imposed by the digital pulses shaping filter circuitry.

4. The integrated circuit synchronous read channel of claim 2 wherein the digital pulse shaping filter circuitry includes variable filter parameters.

5. The integrated circuit synchronous read channel of claim 2 wherein the digital pulse shaping filter circuitry modifies the digitized read signals using programmable filter parameters.

6. The integrated circuit synchronous read channel of claim 1 further comprising the spectrum smoothing filter circuitry for filtering the digitized

read signals prior to processing by the sequence detector.

7. The integrated circuit synchronous read channel of claim 1 wherein the sequence detector processes two digitized read signals at a time, the two digitized read signals respectively representing digitized samples of a read signal of a magnetic storage device during two successive channel bit times.

8. An integrated circuit synchronous read channel for receiving read signals responsive to transitions in magnetic polarity stored in a magnetic storage device and recovering digital data represented thereby comprising:

timing recovery circuitry responsive to the read signals to provide a timing control signal for recovery of the binary digital signal timing in the read signal; and,

a sequence detector responsive to the read signals for receiving a read signal and determining the corresponding sequence of binary digital signals likely to be represented thereby said binary digital signals having a minimum run length constraint of d>0 and said sequence detector comprising a path memory means in which sequences of binary digital signals are constructed, a comparison means to compare the received read signal to a set of ideal read signals, and a selection means which selects an ideal read signal which resembles the received read signal and thereby also selects the sequence of binary digital signals in the path memory which corresponds to the selected ideal read signal.

9. The integrated circuit synchronous read channel of claim 8 wherein d=1.

10. The integrated circuit synchronous read channel of claim 8 wherein said timing recovery circuitry comprises:

timing error measurement circuitry for providing a timing error measurement signal responsive to the present quantity of bit timing error;

timing correction circuitry for correcting the present timing in response to the timing error measurement signal received from the timing error measurement circuitry; and,

sensing circuitry for determining when the timing correction circuitry responds to the timing error measurement signal.

11. The integrated circuit synchronous read channel of claim 10 wherein the sensing circuitry is a transition detector for detecting pulses in the amplitude of the received read signal indicative of magnetic transitions on the storage media.

12. The integrated circuit synchronous read channel of claim 10 further comprised of control circuitry for selecting either the sequence of binary digital signals from the sequence detector or an output from the sensing circuitry.

13. The integrated circuit synchronous read channel of claim 8 further comprised of an RLL(d,k) decoder for providing a run length limited decoded output by decoding the sequence of binary digital signals from the sequence detector.

14. The integrated circuit synchronous read channel of claim 13 wherein the RLL(d,k) decoder is an RLL(1,7) decoder.

15. The integrated circuit synchronous read channel of claim 8 wherein the timing recovery circuitry includes a digital phase detector and a digital timing loop filter.

16. The integrated circuit synchronous read channel of claim 10 wherein the timing error measurement circuitry comprises a digital phase detector and the timing correction circuitry comprises a digital timing loop filter.

17. The integrated circuit synchronous read channel of either of claims 15 and 16 wherein the timing correction circuitry provides a digital frequency control signal responsive to bit timing phase errors for direct control of the frequency of a clock determining bit timing.

18. The integrated circuit synchronous read channel of either of claims 15 and 16 wherein the coefficients of the digital timing loop filter are programmable.

19. The integrated circuit synchronous read channel of claim 8 further comprised of automatic gain control circuitry coupled to the read signals to provide a gain control signal responsive thereto.

20. The integrated circuit synchronous read channel of claim 19 wherein the read signals are digitized read signals representing digitized samples of the read signal of the magnetic storage device and wherein the automatic gain control circuitry includes a digital gain error detector responsive to the digitized read signal samples and a digital gain loop filter.

21. The integrated circuit synchronous read channel of claim 20 wherein the digital gain loop filter provides a digital gain control signal for directly controlling the gain of a variable gain amplifier.

22. The integrated circuit synchronous read channel of claim 20 wherein the coefficients of the digital gain loop filter are programmable.

23. The integrated circuit synchronous read channel of claim 20 wherein the digital gain error detector provides a digital gain error measurement signal responsive to the difference between a programmable desired signal level and the digitized read signal.

24. The integrated circuit synchronous read channel of claim 8 further comprised of digital pulse shaping filter circuitry for modification of the digitized read signals prior to receipt thereof by at least one of (i) the sequence detector and (ii) the timing recovery circuitry.

25. The integrated circuit synchronous read channel of claim 11 further comprised of digital pulse shaping filter circuitry for modification of the digitized read signals prior to receipt thereof by at least one of (i) the sequence detector, (ii) the timing recovery circuitry, and (iii) the transition detector.

26. The integrated circuit synchronous read channel of claim 25 further comprised of delay circuitry for delaying the coupling of the digitized read signals to the transition detector or the timing recovery circuitry to match the delay of the coupling of the

digitized read signals to the timing recovery circuitry or the transition detector, respectively, imposed by the digital pulse shaping filter.

27. The integrated circuit synchronous read channel of either of claims 24 and 25 wherein the digital pulse shaping filter circuitry includes programmable filter parameters.

28. The integrated circuit synchronous read channel of claim 8 further comprised of spectrum smoothing filter circuitry for filtering the digitized read signals to reduce the effect of head bumps prior to processing by the sequence detector.

29. The integrated circuit synchronous read channel of claim 28 wherein the spectrum smoothing filter includes programmable coefficients.

30. The integrated circuit synchronous read channel of claim 28 wherein the spectrum smoothing filter includes programmable delays.

31. The integrated circuit synchronous read channel of claim 8 wherein the read signals are digitized read signals representing digitized samples of the read signal of the magnetic storage device and wherein the sequence detector processes two digitized read signals at a time, the two digitized read signals representing digitized samples of a read signal of a magnetic storage device during two successive channel bit times.

32. The integrated circuit synchronous read channel of claim 8 wherein the read signals are digitized read signals representing digitized samples of

the read signal of the magnetic storage device and wherein the timing recovery circuitry processes two digitized read signals at a time, the two digitized read signals representing digitized samples of a read signal of a magnetic storage device during two successive channel bit times.

33. The integrated circuit synchronous read channel of claim 8 wherein the read signals are analog signals and the read channel includes sampling circuitry to periodically sample the analog signals.

34. The integrated circuit synchronous read channel of claim 33 wherein the sampling circuitry includes digitizing circuitry to produce digitized read signals representing digitized samples of the read signal of the magnetic storage device.

35. The integrated circuit synchronous read channel of claim 8 further comprised of a microprocessor interface and a plurality of control registers.

36. A digital integrated circuit for receiving digitized read signals representing digitized samples of a read signal of a magnetic storage device and recovering digital data represented thereby comprising:

timing recovery circuitry responsive to the digitized read signals to provide a timing control signal for controlling the timing of digitized samples of the read signal; and,

a sequence detector responsive to the digitized read signals for receiving a read signal and determining a corresponding sequence of binary digital signals likely to be represented thereby said binary digital signals having a minimum run length constraint of d>0 and said sequence detector comprising a path memory means in which sequences of binary digital signals are constructed, a comparison means to compare the received read signal to a set of ideal read signals, and a selection means which selects an ideal read signal which resembles the received read signal and thereby also selects the sequence of binary digital signals in the path memory which corresponds to the selected ideal read signal.

37. The digital integrated circuit of claim 36 wherein the timing recovery circuitry is programmable to control the phase of the digitized read signal such that the pulses in the digitized read signal may be selectively side sampled or center sampled.

38. The digital integrated circuit of claim 37 wherein the sequence detector is programmable to determine a sequence of binary digital signals from a digitized read signal in which the pulses are selectively side sampled or center sampled.

39. The digital integrated circuit of claim 36 wherein d=1.

40. The digital integrated circuit of claim 36 wherein said timing recovery circuitry comprises:

timing error measurement circuitry for providing a timing error measurement signal responsive to the present quantity of bit timing error;

timing correction circuitry for correcting the present timing in response to the timing error measurement signal received from the timing error measurement circuitry; and,

sensing circuitry for determining when the timing correction circuitry responds to the timing error measurement signal.

41. The digital integrated circuit of claim 40 wherein the sensing circuitry is a transition detector for detecting pulses in the amplitude of the digitized read signal indicative of magnetic transitions on the storage media.

42. The digital integrated circuit of claim 36 wherein said timing recovery circuitry comprises:

timing error measurement circuitry for providing a timing error measurement signal responsive to the present quantity of bit timing error;

timing correction circuitry for correcting the present timing in response to the timing error measurement signal received from the timing error measurement circuitry; and,

control circuitry responsive to an external transition detector for determining when the timing correction circuitry responds to the timing error measurement signal.

43. The digital integrated circuit of claim 40 further comprised of controllable means for selecting either the sequence of binary digital signals from the sequence detector or the output from the transition detector.

44. The digital integrated circuit of claim 36 further comprised of an RLL(d,k) decoder for providing a run length limited decoded output by decoding the sequence of binary digital signals from the sequence detector.

45. The digital integrated circuit of claim 44 wherein the RLL(d,k) decoder is an RLL(1,7) decoder.

46. The digital integrated circuit of claim 36 wherein the timing recovery circuitry includes a digital phase detector and a digital timing loop filter.

47. The digital integrated circuit of claim 40 wherein the timing error measurement circuitry comprises a digital phase detector and the timing correction circuitry comprises a digital timing loop filter.

48. The digital integrated circuit of either of claims 46 and 47 wherein the timing recovery circuitry provides a digital frequency control signal responsive to bit timing phase errors for direct control of the frequency of a clock determining bit timing.

49. The digital integrated circuit of either of claims 46 and 47 wherein the coefficients of the digital timing loop filter are programmable.

50. The digital integrated circuit of claim 36 further comprised of automatic gain control circuitry coupled to the read signals to provide a gain control signal responsive thereto.

51. The digital integrated circuit of claim 50 wherein the read signals are digitized read signals representing digitized samples of the read signal of the magnetic storage device and wherein the automatic gain control circuitry includes a digital gain error detector responsive to the digitized read signal samples and a digital gain loop filter.

52. The digital integrated circuit of claim 51 wherein the digital gain loop filter provides a digital gain control signal for directly controlling the gain of a variable gain amplifier.

53. The digital integrated circuit of claim 51 wherein the coefficients of the digital gain loop filter are programmable.

54. The digital integrated circuit of claim 36 further comprised of digital pulse shaping filter circuitry for modification of the digitized read signals prior to receipt thereof by at least one of (i) the sequence detector and (ii) the timing recovery circuitry.

55. The digital integrated circuit of claim 41 further comprised of digital pulse shaping filter circuitry for modification of the digitized read signals prior to receipt thereof by at least one of (i) the sequence detector, (ii) the timing recovery circuitry, and (iii) the transition detector.

56. The digital integrated circuit of claim 55 further comprised of delay circuitry for delaying the coupling of the digitized read signals to the transition detector or the timing recovery circuitry to match the delay of the coupling of the digitized read signals to the timing recovery circuitry or the transition detector, respectively, imposed by the digital pulse shaping filter.

57. The digital integrated circuit of either of claims 54 and 55 wherein the digital pulse shaping filter circuitry includes programmable filter parameters.

58. The digital integrated circuit of claim 36 further comprised of spectrum smoothing filter circuitry for filtering the digitized read signals to reduce the

effect of head bumps prior to processing by the sequence detector.

59. The digital integrated circuit of claim 58 wherein the spectrum smoothing filter includes programmable coefficients.

60. The digital integrated circuit of claim 58 wherein the spectrum smoothing filter includes programmable delays.

61. The digital integrated circuit of claim 36 wherein the read signals are digitized read signals representing digitized samples of the read signal of the magnetic storage device and wherein the sequence detector processes two digitized read signals at a time, the two digitized read signals representing digitized samples of a read signal of a magnetic storage device during two successive channel bit times.

62. The digital integrated circuit of claim 36 wherein the read signals are digitized read signals representing digitized samples of the read signal of the magnetic storage device and wherein the timing recovery circuitry processes two digitized read signals at a time, the two digitized read signals representing digitized samples of a read signal of a magnetic storage device during two successive channel bit times.

63. The digital integrated circuit of claim 36 further comprised of an RLL encoder for RLL encoding digital data to be written to a magnetic storage device.

64. The digital integrated circuit of claim 36 further comprised of:

a processor interface and a plurality of control registers from which a processor may read information and to which a processor may write information; and,

a serial interface having a serial data line and a serial clock line.

65. The digital integrated circuit of claim 64 further comprised of:

multiplexing circuitry for providing as the output of the integrated circuit;

i) the digital data to be written to a magnetic storage device in 2 bit wide form during writing to the storage medium; and,

ii) selectively providing the output of the serial data line and the serial clock line when not writing to the storage medium.

66. The digital integrated circuit of claim 64 wherein the serial interface includes circuitry for reading in serial form on the serial data line the contents of mapped external registers.

67. The digital integrated circuit of claim 66 wherein the serial interface includes circuitry responsive to requests presented at the processor interface to selectively:

a) read in serial form on the serial data line the contents of one of the mapped external registers and then provide the contents of said mapped external register at the processor interface; or

b) provide at the processor interface the contents of a mapped external register specified in a previous request and then read in serial form on the serial data line the contents of a second mapped external register specified in the present request.

68. The digital integrated circuit of claim 66 wherein a processor may write control information to a specified mapped external register by writing to the mapped external register's address in the digital integrated circuit, and wherein the serial interface includes circuitry for presenting in serial form on the serial data line to the specified mapped external register the data written to the mapped external register's address in the digital integrated circuit each time the processor writes to the mapped external register's address.

69. A digital integrated circuit comprising:

an RLL encoder for RLL encoding digital data to be written to a magnetic storage device;

a processor interface and a plurality of control registers from which a processor may read information and to which a processor may write information; and,

a serial interface having a serial data line and a serial clock line;

multiplexing circuitry for providing as the output of the integrated circuit

i) the output of the RLL encoder in 2 bit wide form during writing to the storage medium; and,

ii) selectively providing the output of the serial data line and the serial clock line when not writing to the storage medium.

70. The digital integrated circuit of claim 69 wherein the serial interface includes circuitry for reading in serial form on the serial data line the contents of mapped external registers.

71. The digital integrated circuit of claim 70 wherein the serial interface includes circuitry

responsive to requests presented at the processor interface to selectively:

a) read in serial form on the serial data line the contents of one of the mapped external registers and then provide the contents of said mapped external register at the processor interface; or

b) provide at the processor interface the contents of a mapped external register specified in a previous request and then read in serial form on the serial data line the contents of a second mapped external register specified in the present request.

72. The digital integrated circuit of claim 69 wherein a processor may write control information to a specified mapped external register by writing to the mapped external register's address in the digital integrated circuit, and wherein the serial interface includes circuitry for presenting in serial form on the serial data line to the specified mapped external register the data written to the mapped external register's address in the digital integrated circuit each time the processor writes to the mapped external register's address.

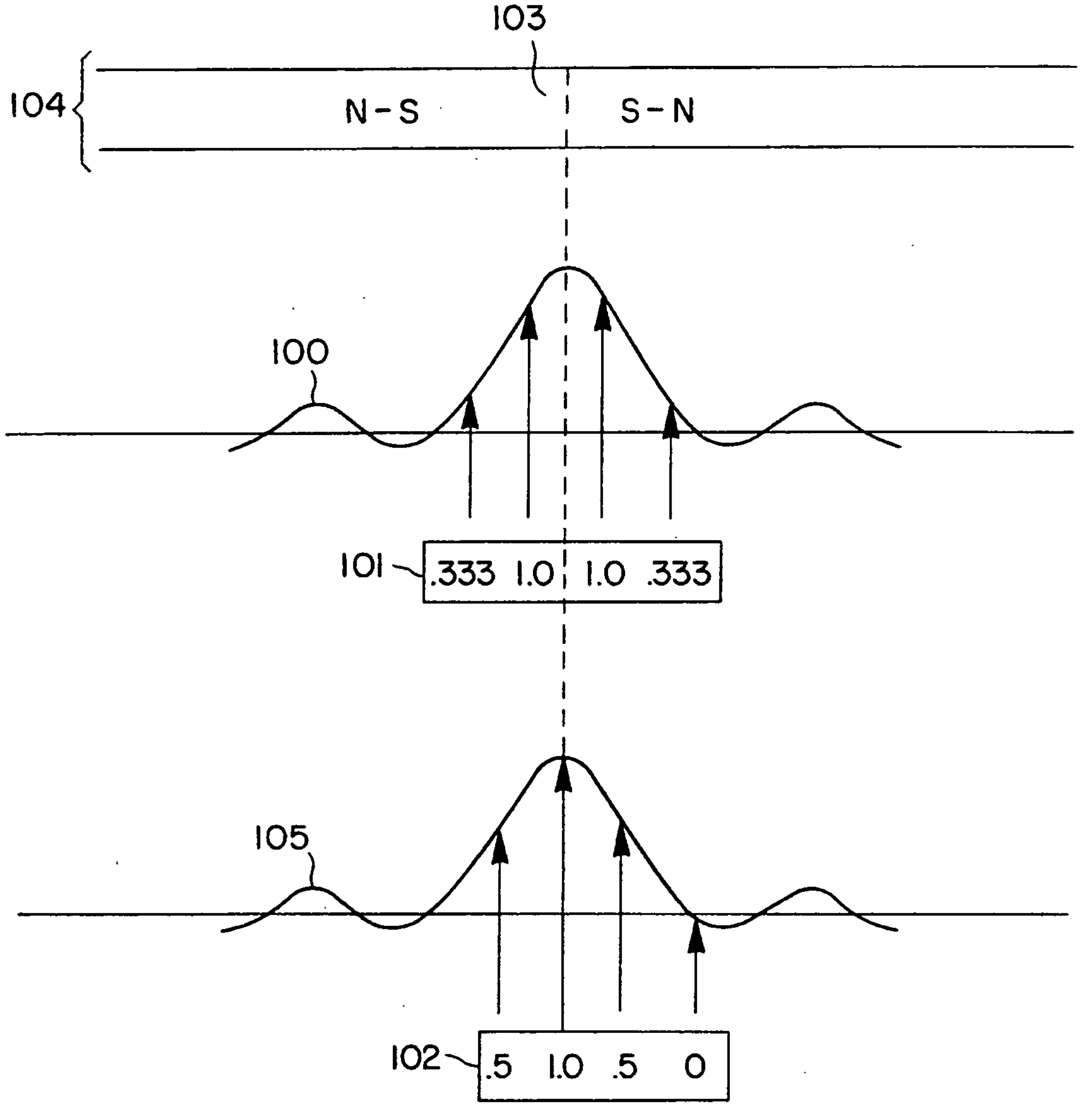


FIG. 1
(PRIOR ART)

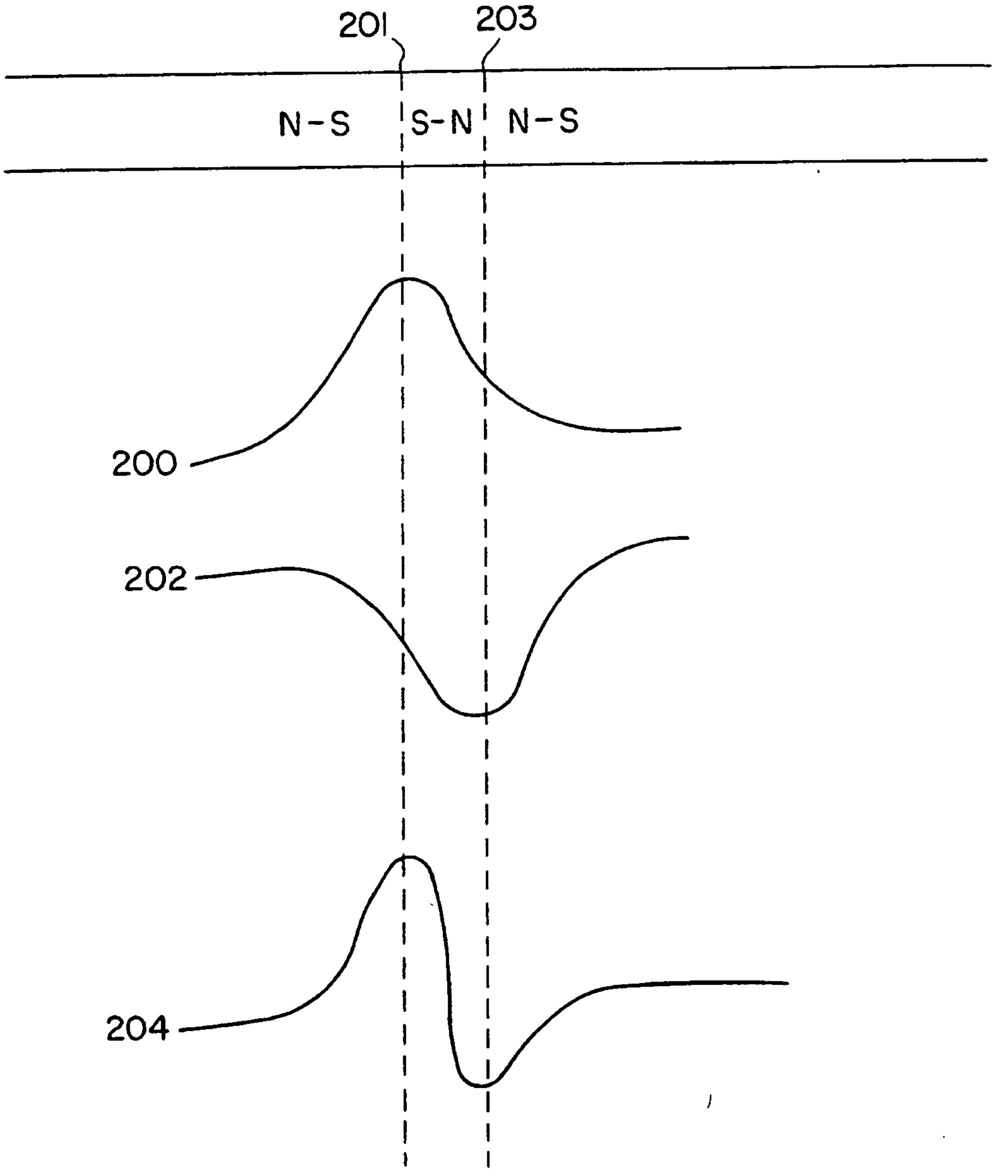


FIG. 2
(PRIOR ART)

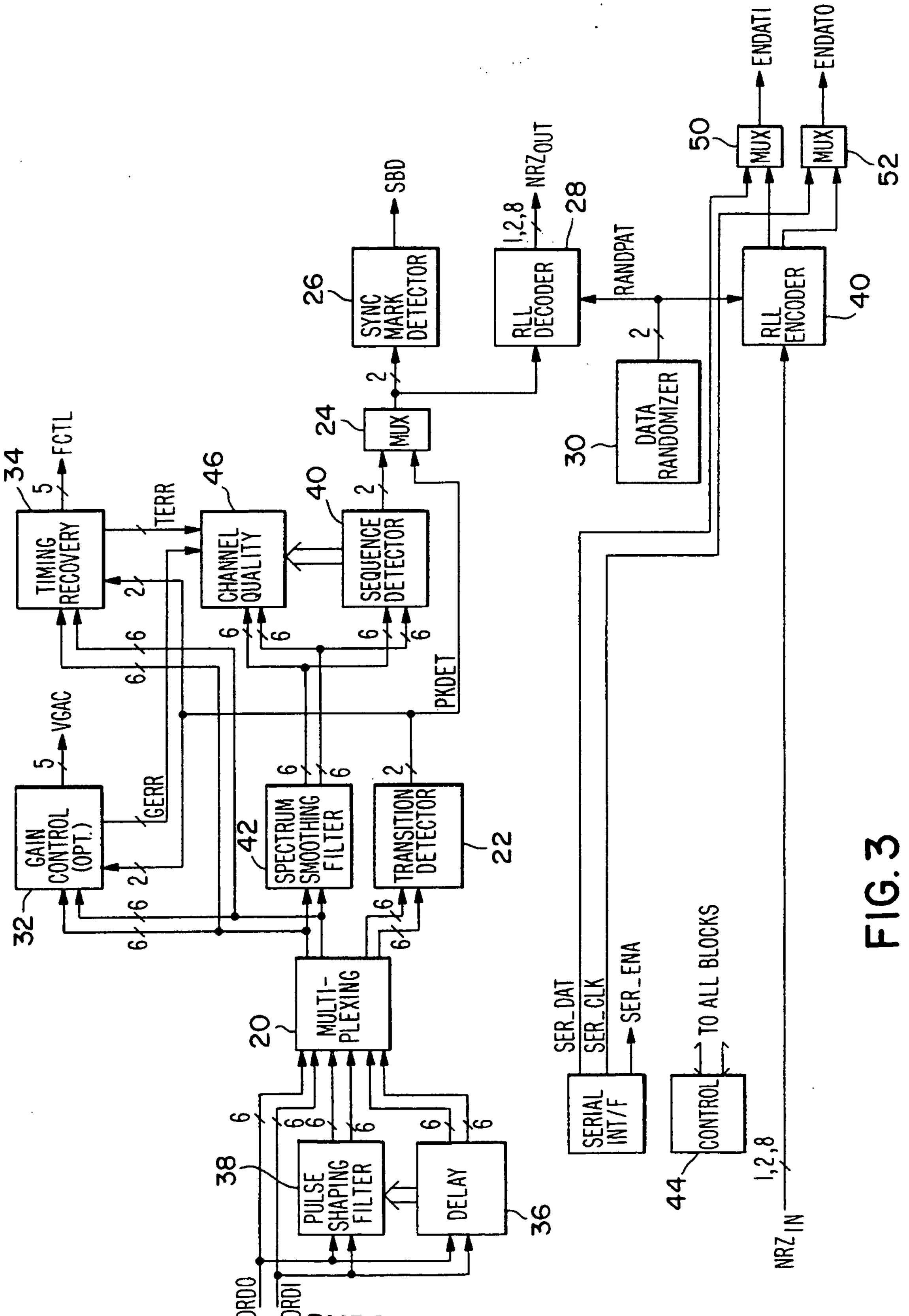


FIG. 3

SUBSTITUTE SHEET

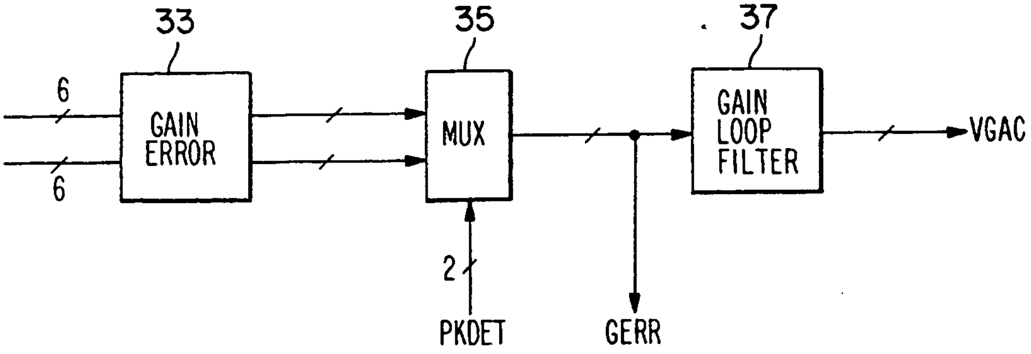


FIG. 4

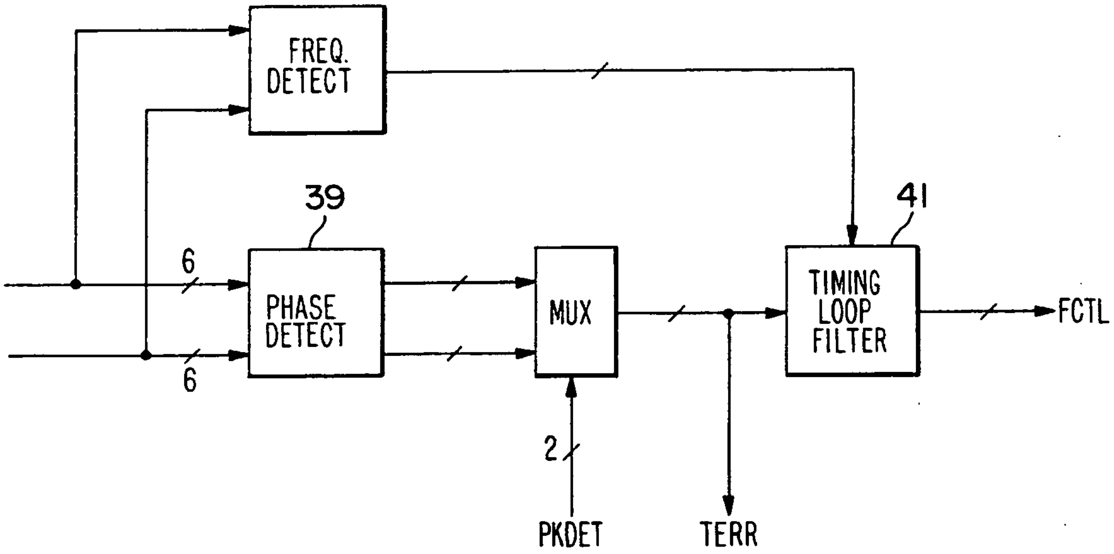


FIG. 5

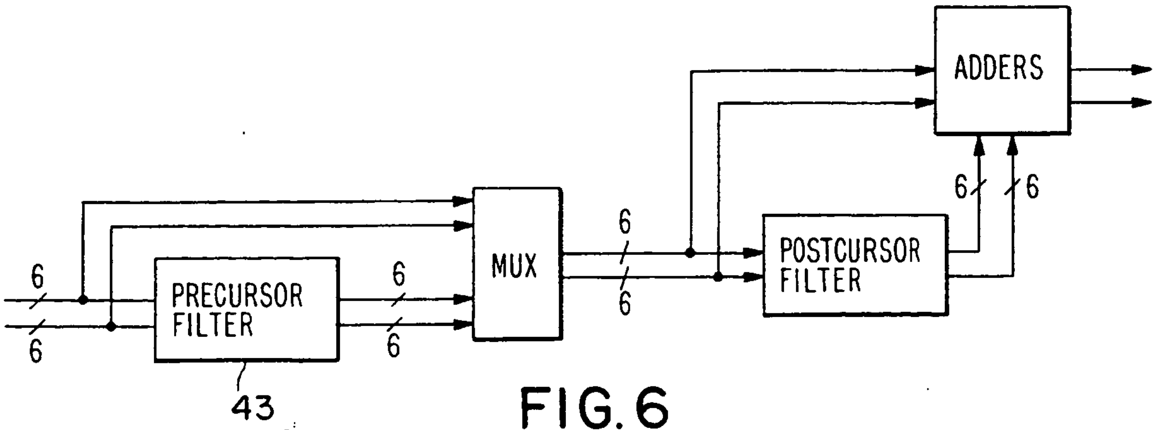


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/01084

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :Please See Extra Sheet.
US CL : 360/40, 46
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 360/31, 32, 39, 40, 46, 48, 49, 51, 53, 59; 341/59

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS
search terms: RLL, sequence detector, peak detector, timing recovery

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US, A, 4, 945, 538 (Patel) 31 July 1990.	1-7
A	I.E.E.E. Journal on Selected Areas in Communications, Volume 27, No. 6, issued January 1992, R.D. Cideciyan et al., "A PRML System for Digital Magnetic Recording", pages 38-56.	1-7
A	I.E.E.E. International Solid-State Circuits Conference, 1991, T.J. Schmerback et al., "A 27 MHz Mixed Analog/Digital Magnetic Recording Channel DSP Using Partial Response Signaling with Maximum Likelihood Detection", pages 136-137.	1-7

☒ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
06 May 1994

Date of mailing of the international search report
MAY 23 1994

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. NOT APPLICABLE

Authorized officer
PATRICK WAMSLEY
Telephone No. (703) 308-0956

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/01084

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	IBM Storage Systems Products Division, Volume 27, No. 6, 7 July 1991, J.D. Coker et al ., "Implementation of PRML in a Rigid Disk Drive".	1-7
A	IBM Storage Systems Products Division, Volume 27, No. 6, November 1991, J. Hong et al., "An Experimental 180 Mb/sec PRML Channel for Magnetic Recording".	1-7
A, P	US, A, 5,291,499 (Behrens et al.)01 March 1994.	1-72
A, P	US, A, 5,297,184 (Behrens et al.) 22 March 1994.	1-72
A, E	US, A, 5,268,908 (Glover et al.)07 December 1993.	1-72.
A	US, A, 4,839,896, (Glover et al.) 13 June 1989.	1-72

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/01084

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. □ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. □ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. □ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

I. Claims 1-68, drawn to timing recovery and sequence detection, classified in 360/40.
II. Claims 69-72, drawn to interfaces and multiplexing circuitry, classified in 360/46.

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. □ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. □ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. □ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest □ The additional search fees were accompanied by the applicant's protest.
☒ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/01084

A. CLASSIFICATION OF SUBJECT MATTER:
IPC (5):

G11B 5/09, 20/14, 20/16